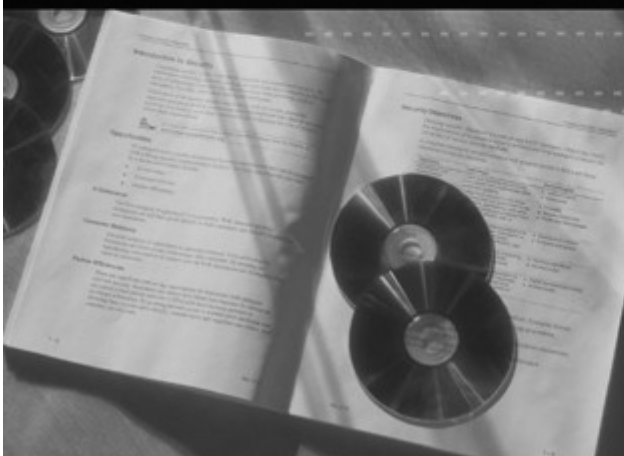


*OpenVMS System and Network Node
Management II*

Instructor Guide

Training

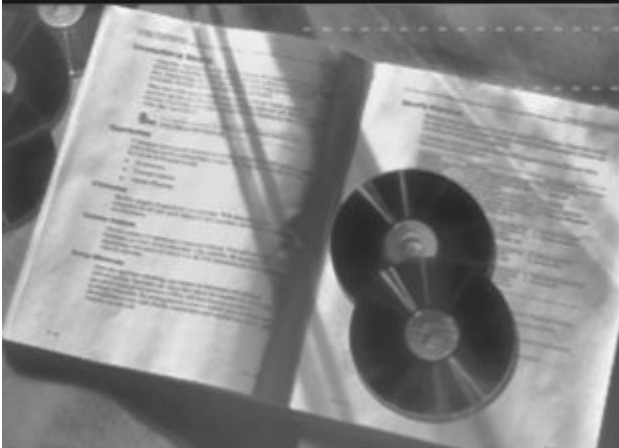


COMPAQ

***OpenVMS System and Network Node
Management II***

Instructor Guide

Training



COMPAQ

Notice

The information in this publication is subject to change without notice.

COMPAQ COMPUTER CORPORATION SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL.

This guide contains information protected by copyright. No part of this guide may be photocopied or reproduced in any form without prior written consent from Compaq Computer Corporation.

The software described in this guide is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of this agreement.

Other product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

©2000 Compaq Computer Corporation. All rights reserved. Printed in the USA.

Aero, ALPHA, ALPHA AXP, AlphaServer, AlphaStation, Armada, BackPaq, COMPAQ, Compaq Insight Manager, CompaqCare logo, Counselor, DECterm, Deskpro, DIGITAL, DIGITAL logo, DIGITAL Alpha Systems, Digital Equipment Corporation, DIGITAL UNIX, DirectPlus, FASTART, Himalaya, InfoPaq, Integrity, LicensePaq, Ministation, NetFlex, NonStop, OpenVMS, PaqFax, Presario, ProLiant, ProLinea, ProSignia, QuickBack, QuickFind, Qvision, RDF, RemotePaq, RomPaq, ServerNet, SERVICenter, SmartQ, SmartStart, SmartStation, SolutionPaq, SpeedPaq, StorageWorks, Systempro/LT, Tandem, TechPaq, TruCluster, and Tru64 UNIX are registered in United States Patent and Trademark Office.

Atalla, C-Series, Expand, FOX, Guardian, iTP, Measure, Netelligent, and PointView are trademarks of Compaq Computer Corporation.

Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Intel, Pentium, and Intel Inside are registered trademarks and Xeon is a trademark of Intel Corporation.

UNIX is a trademark in the US and other countries, licensed exclusively through X-Open Company Ltd.

OpenVMS System and Network Node Management II

Instructor Guide

January 2000

Overview

About This Course.....	Overview-1
Introduction.....	Overview-1
Topics.....	Overview-1
Course Description	Overview-1
Prerequisites	Overview-2
Resources	Overview-3
Course Organization	Overview-4
Course Goals	Overview-5
Course Nongoes	Overview-5
Course Conventions.....	Overview-6
Materials.....	Overview-7
Introduction.....	Overview-7
Student Package	Overview-7
Instructor Package	Overview-8
Transparencies.....	Overview-8
Using Tables.....	Overview-8
Hardware Needed	Overview-8
Preparing to Teach.....	Overview-9
Introduction.....	Overview-9
Preparing the System.....	Overview-9
Selecting Equipment.....	Overview-9
Creating Student Accounts.....	Overview-10

Module 1 — Understanding System Resources

About This Module.....	1-2
Introduction.....	1-2
Objectives.....	1-2
Virtual Memory Management Concepts.....	1-3
Overview	1-3
Topics.....	1-3
Virtual and Physical Memory.....	1-4
Virtual Address Space	1-5
Physical Address Space	1-6
Virtual Address Space Layout.....	1-7
Layout of Virtual Address Space	1-7
Virtual Address Space Mapped to Physical Memory.....	1-8
Working Sets.....	1-9
Overview	1-9
Topics.....	1-9

Working Set Concepts	1-10
Working Set Limit	1-10
Parameters Controlling Working Set.....	1-10
Working Set Parameters.....	1-11
Example: Working Set Parameters for Interactive Processes	1-11
Example Notes: Working Set Parameters for Interactive Processes	1-12
Working Set Parameters for Batch Processors.....	1-12
Example: Displaying Queue Parameters	1-13
Working Set Parameters for Other Processes	1-14
Default Working Set Values	1-14
Paging and Swapping.....	1-15
Overview	1-15
Topics.....	1-15
Paging	1-16
Page Faults	1-16
Image Startup	1-17
Page Table.....	1-18
Page Table Entry	1-18
Global Pages.....	1-18
Page Fault Types	1-19
Hard Faults	1-19
Removing a Page from the Working Set	1-20
Free Page List.....	1-20
Modified Page List	1-22
Page Fault Read Clustering.....	1-23
Swapping.....	1-24
SWAPPER Process.....	1-24
Scheduling Resources	1-25
Overview	1-25
Topics.....	1-25
Interrupts	1-26
Methods of Scheduling	1-26
Interrupting.....	1-26
Interrupt Priority Levels.....	1-27
Types of Priority.....	1-27
Interrupt Priority Level	1-27
IPL and Software Priority	1-28
Scheduler.....	1-29
Time Slice	1-29
Process Selection Criteria	1-29
Invoking the Scheduler	1-29
Scheduling on a Multiprocessor System.....	1-30

Processor Affinity Capability	1-31
Use of the Qualifiers	1-31
Qualifiers for SET PROCESS Command	1-31
/CAPABILITIES Qualifier	1-31
/AFFINITY Qualifier	1-33
Secondary Qualifiers for /AFFINITY	1-34
Qualifier for START/CPU	1-35
Display for SHOW CPU Command	1-35
Scheduling Priority	1-36
Base Priority	1-36
Current Priority	1-36
Process States	1-38
Voluntary Wait States	1-38
Involuntary Wait State	1-38
Process Behavior	1-39
Figure: Process State Transition	1-39
Dynamic Priority Adjustment	1-40
Priority Boosting	1-40
System Events and Associated Priority Boosts	1-40
Priority Decrementing	1-41
Results of Priority Adjustment	1-41
Example: Displaying Process States and Priorities	1-42
Network Configurations	1-43
Overview	1-43
Topics	1-43
Network Configurations	1-44
DECnet Network	1-44
DECnet Network Characteristics	1-44
DECnet Network Configurations	1-45
DECnet Node Names	1-45
Full Names Syntax	1-46
Example: Full Names Syntax	1-46
Rules for Assigning Full Names	1-47
Network Hardware	1-48
Point-to-Point Connections	1-48
Figure: Point-to-Point Connection	1-48
FDDI LAN Standard	1-49
Figure: FDDI Network	1-49
802.3/Ethernet LAN Standard	1-50
Figure: Ethernet LAN	1-52
Figure Notes: Ethernet LAN	1-53
Servers	1-54
Extended LANs	1-55
Figure: Extended LAN	1-56
Multivendor Networks	1-57
Figure: DECnet Software in a Multivendor Environment	1-57

DECnet Routing	1-58
Routing.....	1-58
How Routing Works.....	1-58
Routing Terminology.....	1-59
Node Types	1-60
Area Routing	1-60
Figure: DECnet Node Types.....	1-62
Routing on an Ethernet LAN	1-62
Summary	1-64
Concepts.....	1-64
Virtual Memory Management Concepts	1-64
Working Sets.....	1-64
Paging and Swapping	1-64
Scheduling Resources.....	1-65
Network Configurations.....	1-65

Module 2 — Managing Batch and Print Requests

About This Module.....	2-2
Introduction.....	2-2
Objectives.....	2-2
Managing Queues and Spooled Devices.....	2-3
Overview	2-3
Topics.....	2-3
OpenVMS Management Station.....	2-4
Printer Management	2-4
Printer Management Operations.....	2-5
Creating Execution Queues.....	2-6
Execution Queues	2-6
Creating an Output Execution Queue	2-6
Example: Creating an Output Execution Queue	2-6
Example Notes: Creating an Output Execution Queue	2-6
Creating Generic Queues	2-7
Generic Queues	2-7
Creating a Generic Batch Queue.....	2-7
Sample Commands for Creating Generic Batch Queues.....	2-8
Creating Generic Print Queues.....	2-8
Deleting Queues	2-9
Deleting Queues	2-9
Example: Deleting a Queue	2-9
Example Notes: Deleting a Queue.....	2-9
Naming Queues	2-10
Queue Names	2-10
Naming a Queue Using the Default Value.....	2-10
Naming a Print Execution Queue.....	2-10

Managing Queue Characteristics.....	2-11
Queue Characteristics	2-11
Defining Characteristics.....	2-12
Example: Defining Queue Characteristics.....	2-12
Example: Displaying Characteristics Defined on a System.....	2-12
Assigning Characteristics to a Queue	2-13
Example Assigning Characteristics to a Queue	2-13
Example: Displaying Characteristics Assigned to a Queue.....	2-13
Example Notes: Displaying Characteristics Assigned to a Queue.....	2-13
Canceling Characteristics Assigned to a Queue.....	2-14
Deleting Characteristics	2-14
Creating Batch Queues	2-15
Batch Processing	2-15
Creating a Batch Queue	2-15
Example: Creating a Batch Queue.....	2-16
Example Notes: Creating a Batch Queue.....	2-16
Creating Print Queues.....	2-17
Print Queues	2-17
Setting Device Characteristics for Printers	2-17
Example: Setting up Device Characteristics	2-17
Example Notes: Setting Up Device Characteristics	2-17
Creating an Output Execution Queue	2-18
Creating Generic Print Queues.....	2-18
Creating Additional Queue Managers	2-19
Additional Queue Managers.....	2-19
Creating Additional Queue Managers	2-19
Example: Creating Additional Queue Managers.....	2-20
Example Notes: Creating Additional Queue Managers.....	2-20
Creating Queues Under Multiple Queue Managers.....	2-21
Queues Under Multiple Queue Managers.....	2-21
Example: Creating Queues Under Multiple Queue Managers.....	2-22
Example Notes: Creating Queues Under Multiple Queue Managers.....	2-22
Setting Up Logical Ports.....	2-23
Logical Ports	2-23
Setting Up Logical Ports for LAT Printers	2-23
Example: Setting Device Characteristics for a Logical Port	2-23
Example Notes: Setting Device Characteristics for a Logical Port.....	2-23
Setting Up Spooled Devices.....	2-24
Spooling	2-24
Creating a Spooled Output Device	2-24
Example: Setting Up Spooled Output devices	2-25
Example Notes: Setting Up Spooled Output Devices	2-25
Changing Device Characteristics on a Spooled Device.....	2-25
Controlling Batch and Print Jobs.....	2-26
Overview	2-26
Topics.....	2-26

Controlling Batch Queue Execution.....	2-27
Limiting the Number of Jobs	2-27
Establishing the Scheduling Priority for Batch Jobs	2-27
Controlling Batch Process Working Sets.....	2-28
Using Device Control Libraries.....	2-29
Device Control Libraries.....	2-29
Types of Device Control Modules.....	2-29
Steps for Using Device Control Libraries.....	2-29
Creating a Device Control Library	2-29
Determining the Contents of a Device Control Library.....	2-30
Creating a Device Control Module.....	2-30
Inserting a Module into a Device Control Library	2-30
Assigning a Library to a Queue.....	2-30
Listing the Modules in a Library.....	2-31
User Access to Device Control Modules	2-31
Creating Forms for Setup and Page Setup Modules.....	2-31
Assigning a Reset Module to a Queue.....	2-32
Example: Using Device Control Libraries.....	2-33
Example Notes: Using Device Control Libraries.....	2-33
Saving and Restoring the Queue Database	2-34
Saving the Queue Database.....	2-34
Steps for Saving the Queue Database	2-34
Example: Saving the Queue Database	2-35
Example Notes: Saving the Queue Database	2-35
Restoring the Queue Database	2-36
Steps for Restoring the Queue Database.....	2-36
Example: Restoring the Queue Database.....	2-37
Example Notes: Restoring the Queue Database.....	2-37
Using Command Procedures to Automate Queue Creation.....	2-38
Automating Queue Creation.....	2-38
Device Characteristics	2-38
Spooling Devices.....	2-38
Automatic Startup at System Initialization	2-38
Generic Queue Startup.....	2-39
Using Forms	2-40
Forms	2-40
Managing Forms.....	2-40
Defining a Form	2-41
Specifying a Form	2-41
Example: Defining a Form.....	2-42
Example Notes: Defining a Form.....	2-42
Providing Formatting Options Only	2-42
Example: Displaying Form Information.....	2-42
Assigning a Default Form for a Queue.....	2-43
Example: Displaying a Default Form for a Queue.....	2-43

Example Notes: Displaying a Default Form for a Queue	2-43
Mounted Forms	2-44
Changing the Mounted Form	2-44
Deleting a Form.....	2-44
Summary	2-45
Concepts.....	2-45
Managing Queues and Spooled Devices	2-45
Controlling Batch and Print Jobs.....	2-46
Commands.....	2-47
Managing Queues and Spooled Devices	2-47
Controlling Batch and Print Jobs.....	2-48

Module 3 — Managing LAT Services

About This Lesson.....	3-2
Introduction.....	3-2
Objectives.....	3-2
LAT Network Concepts.....	3-3
Overview	3-3
Topics.....	3-3
Local Area Transport.....	3-4
LAT Network.....	3-4
LAT Protocol.....	3-4
Figure: LAT Network.....	3-5
Service Nodes.....	3-6
Session	3-6
Service Advertisements	3-6
Types of Service Requests	3-7
Terminal Servers.....	3-8
What Is a Terminal Server?.....	3-8
Terminal Server Operating Modes	3-8
Permanent Terminal Server Database.....	3-9
Operational or Volatile Terminal Server Database	3-9
Displaying Server Characteristics	3-10
Testing a Terminal Server Port	3-10
Example: Testing a Terminal Server Port.....	3-11
Using the LAT Control Program.....	3-12
Overview	3-12
Topics.....	3-12
LAT Control Program.....	3-13
LATCP	3-13
Functions of LATCP.....	3-13
Running LATCP.....	3-13
Sample LATCP Commands.....	3-13
Setting Up a LAT Cluster Alias	3-14
Overview	3-14
Topics.....	3-14

LAT Cluster Alias.....	3-15
Creating a LAT Cluster Alias at System Startup	3-15
Creating a LAT Cluster Alias from DCL	3-15
Example: Creating a LAT Cluster Alias from DCL.....	3-15
Setting Up a Remote Print Queue	3-16
Overview	3-16
Topics.....	3-16
Initial Actions	3-17
Initial Actions	3-17
Verifying Printer Settings	3-17
Example: Displaying Port Characteristics	3-17
Step 1: Set Up the Terminal Server	3-18
Steps for Setting Up the Terminal Server	3-18
Install and Initialize the Terminal Server.....	3-18
Select a Terminal Server Name.....	3-18
Set the Terminal Server Name	3-18
Verify the Terminal Server Name	3-18
Example: Verifying the Terminal Server Name.....	3-19
Step2: Set Up the Printer Port on the Server.....	3-20
Steps for Setting Up the Printer Port	3-20
Example: Setting Up a Printer Port	3-21
Step 3: Set Up the Application Port.....	3-22
Steps for Setting Up the Application Port	3-22
Select an Application Port Name	3-22
Create the Application Port.....	3-22
Map the Application Port.....	3-23
Verify the Port Mapping	3-23
Make the Application Port Setup Permanent	3-24
Example: Creating and Mapping an Application Port.....	3-24
Step 4: Set Up the Print Queue.....	3-25
Steps for Setting Up the Print Queue.....	3-25
Set Up the Printer Characteristics.....	3-26
Set the Device Protection.....	3-27
Set the Device Spooled	3-27
Initialize and Start the Queue.....	3-28
Example: Initializing and Starting a Queue	3-29
Test the Printer	3-29
Summary	3-30
Concepts.....	3-30
LAT Network Concepts.....	3-30
Using the LAT Control Program.....	3-30
Setting Up a LAT Cluster Alias	3-30
Setting Up a Remote Print Queue	3-30
Commands.....	3-32
LAT Network Concepts.....	3-32
Using the LAT Control Program.....	3-32
Setting Up a LAT Cluster Alias	3-32
Setting Up a Remote Print Queue	3-32

Module 4 — Managing Volumes

About This Module	4-2
Introduction	4-2
Objectives	4-2
Review of OpenVMS Device Concepts	4-3
Overview	4-3
Topics	4-3
OpenVMS Device Names	4-4
Device Names	4-4
Parts of a Device Name	4-4
Hardware Controller Number	4-4
Unit Number	4-4
Devices in an OpenVMS Cluster System	4-5
MSCP Server	4-5
Tape MSCP Server	4-5
Device Name Formats	4-7
Node\$device Format	4-7
Allocation Class	4-7
Duplicate Device Names	4-7
Naming Devices on a Local Node	4-8
Figure: Sample OpenVMS Cluster System	4-9
SCSI Clusters	4-11
Overview	4-11
Topics	4-11
Interconnect Concepts	4-12
Accessing SCSI Storage	4-12
Multiple-Host SCSI Access	4-12
The SCSI Standard	4-13
Devices	4-13
Performance	4-14
Distance	4-14
Cabling and Termination	4-15
SCSI OpenVMS Cluster Hardware Configurations	4-16
Add-On SCSI Adapters	4-17
Using KZPAA Adapters	4-17
Building a Basic System	4-17
Removing a Server	4-17
HSZ40 Controllers	4-18
Additional Enclosures	4-18
Internal SCSI Adapters	4-19
Benefits	4-19
Limitations	4-19
SCSI Environment Parameters	4-20
Review of System Directories	4-21
Overview	4-21
Topics	4-21

OpenVMS System Directories	4-22
System Directories.....	4-22
Concealed Root Directories	4-22
Figure: System Directory Tree.....	4-23
OpenVMS System Directories and Their Contents.....	4-24
Other System Logical Names.....	4-25
Obtaining Information About Volumes	4-26
Overview	4-26
Topics.....	4-26
Volume Types and Characteristics	4-27
Public Volumes	4-27
Private Volumes	4-27
Volume Label	4-28
Other Information Encoded on a Volume.....	4-28
Physical Volume Labels.....	4-28
Displaying Device Characteristics.....	4-29
SHOW DEVICE Command.....	4-29
Example: SHOW DEVICE Command.....	4-29
Example: Locally Connected Disks.....	4-30
Example Notes: Locally Connected Disks.....	4-30
Example: Remote Disks.....	4-31
Example Notes: Remote Disks.....	4-31
Example: ISO 9660 CD-ROM Disks	4-32
Example Notes: ISO 9660 CD-ROM Disks.....	4-32
Open Files on a Volume	4-34
Listing Open Files	4-34
Example: Listing Open Files.....	4-35
Preparing Volumes for Use.....	4-36
Overview	4-36
Topics.....	4-36
Allocating a Device	4-37
Gaining Exclusive Access to a Device	4-37
Allocating a Specific Device.....	4-37
Allocating a Generic Device	4-38
Initializing a Volume	4-39
Requirements for Initializing a Volume	4-39
INITIALIZE Command.....	4-39
Initializing a Public Disk.....	4-40
Initializing a Private Disk	4-40
Initializing a Group Disk.....	4-40
Changing Protection on Initialization.....	4-40
Initializing a Tape Volume.....	4-41
Mounting a Volume.....	4-42
MOUNT Command	4-42
Mounting a Public Disk	4-42
Mounting a Disk Clusterwide	4-43
Setting Protection on MOUNT	4-43

Removing Volumes	4-44
Dismounting a Volume	4-44
Deallocating a Device	4-44
Creating a Public Volume	4-46
Steps for Creating a Public Volume	4-46
Initializing a Public Volume	4-46
Mounting a Public Volume	4-46
Example: Preparing a Public Disk.....	4-47
Example Notes: Preparing a Public Disk.....	4-47
Creating a Volume Set.....	4-48
Volume Sets	4-48
Procedure for Creating a Volume Set.....	4-48
Dismounting a Volume Set.....	4-48
Example: Creating a Volume Set	4-49
Example Notes: Creating a Volume Set	4-50
Handling Common Volume Problems.....	4-51
Overview	4-51
Topics.....	4-51
Setting File Characteristics.....	4-52
SET FILE Command	4-52
File Characteristics That Can Be Modified.....	4-52
Modifying the Action of the SET FILE Command.....	4-53
Example: Changing File Characteristics.....	4-54
Example Notes: Changing File Characteristics.....	4-55
Defragmenting a Disk.....	4-56
Disk Fragmentation	4-56
Using BACKUP to Defragment a Disk	4-56
Handling Files Marked NOBACKUP	4-56
Example: Using BACKUP to Defragment a Disk	4-56
Rebuilding Incorrectly Dismounted Disks.....	4-57
Incorrectly Dismounted Disks.....	4-57
Automatically Rebuilding a Disk	4-57
Explicitly Rebuilding a Disk	4-57
Delaying a Disk Rebuild Operation.....	4-57
Specifying a New Volume Label.....	4-58
Using the SET VOLUME Command	4-58
Example: Specifying a New Volume Label.....	4-58
Summary	4-59
Concepts.....	4-59
Review of OpenVMS Device Concepts	4-59
SCSI Clusters	4-59
Review of System Directories.....	4-59
Obtaining Information About Volumes.....	4-59
Preparing Volumes for Use.....	4-60
Handling Common Volume Problems.....	4-60

Commands.....	4-61
Obtaining Information About Volumes.....	4-61
Preparing Volumes for Use.....	4-61
Handling Common Volume Problems.....	4-62

Module 5 — Introduction to System Customization

About This Module.....	5-2
Introduction.....	5-2
Objectives.....	5-2
Customizing System Startup Files.....	5-3
Overview.....	5-3
Topics.....	5-3
Optional Files.....	5-4
Adding or Deleting Files.....	5-4
Site-Independent Startup File.....	5-5
STARTUP.COM.....	5-5
Sequence of Operations for Startup.....	5-6
Component Files Used by STARTUP.COM.....	5-8
How STARTUP.COM Starts the System.....	5-8
Do Not Modify Site-Independent Files.....	5-9
Configuring Devices.....	5-10
SYCONFIG.COM.....	5-10
Use of SYCONFIG.COM.....	5-10
Example: SYCONFIG.COM.....	5-10
Defining Systemwide Logical Names.....	5-12
SYLOGICALS.COM.....	5-12
How to Create System Logical Names.....	5-12
How to Delete System Logical Names.....	5-12
Standard Logical Names.....	5-12
Executive-Mode Logical Names.....	5-13
How to Create Executive-Mode Logical Names.....	5-14
Example: Assigning Site-Specific Logical Names.....	5-14
Example: SYLOGICALS.COM.....	5-15
Installing Page and Swap Files.....	5-17
SYPAGSWPFILES.COM.....	5-17
Installing Page and Swap Files.....	5-17
Example: Installing Page and Swap Files.....	5-18
Performing Site-Specific Startup Functions.....	5-19
SYSTARTUP_VMS.COM.....	5-19
Functions of SYSTARTUP_VMS Command Procedure.....	5-19
Using Separate Site-Specific Command Procedures.....	5-19
Example: SYSTARTUP_VMS.COM.....	5-20
Customizing Login Command Procedures.....	5-22
Overview.....	5-22
Topics.....	5-22

Introduction to Login Command Procedures	5-23
Login Command Procedures.....	5-23
Location of Login Command Procedures	5-23
Order of Execution of Login Command Procedures	5-23
Determining Execution Mode	5-23
Uses for Login Command Procedures.....	5-24
Personal Login Command Procedures.....	5-24
System and Personal Login Command Procedures.....	5-24
System and Group Login Command Procedures	5-24
Captive Login Command Procedures.....	5-25
Customizing System Parameters	5-26
Overview	5-26
Topics.....	5-26
System Parameters.....	5-27
Default Parameter Values	5-27
Installed Images.....	5-28
System Files	5-28
Tools for Maintaining System Parameters.....	5-29
System Generation Utility, SYSGEN.....	5-29
AUTOGEN Command Procedure.....	5-29
System Management Utility, SYSMAN.....	5-30
Conversational Startup.....	5-30
Reconfiguring the System with AUTOGEN.....	5-31
AUTOGEN.COM.....	5-31
Functions of AUTOGEN.COM	5-31
When You Should Run AUTOGEN	5-31
When AUTOGEN Changes Take Effect	5-31
Running AUTOGEN	5-32
Privileges Required to Run AUTOGEN.....	5-32
AUTOGEN Phases	5-32
Steps for Changing System Parameters Without Changing File Sizes.....	5-34
Steps for Changing System Parameters and File Sizes.....	5-35
Example: MODPARAMS.DAT.....	5-36
Maintaining Parameters with SYSMAN.....	5-37
SYSMAN	5-37
SYSMAN PARAMETERS Command.....	5-37
SYSMAN PARAMETERS Subcommands	5-37
Displaying Parameters	5-38
Example: Displaying System Parameters	5-39
Example Notes: Displaying System Parameters	5-39
Maintaining the DECnet for OpenVMS Remote Node Database	5-40
Overview	5-40
Topics.....	5-40
DECnet Configuration Database	5-41
Permanent Database.....	5-41
Volatile Database.....	5-41

Using NCP.....	5-42
NCP.....	5-42
Invoking NCP.....	5-42
Exiting from NCP.....	5-42
NCP Commands for Maintaining the Configuration Database.....	5-42
Privileges Needed to Use NCP.....	5-43
Protection on Database Files.....	5-43
Adding Remote Nodes.....	5-44
Identifying Remote Nodes.....	5-44
Guaranteeing Unique Names and Addresses.....	5-44
Strategy for Setting Up Node Names.....	5-44
Example: Setting Up a Node Name.....	5-45
Steps for Adding a Node.....	5-46
Example: Adding a Remote Node.....	5-47
Example Notes: Adding a Remote Node.....	5-48
Procedure for Copying Known Nodes.....	5-48
Changing a Remote Node Entry.....	5-49
Reasons for Changing a Remote Node Entry.....	5-49
How to Change a Remote Node Entry.....	5-49
Example: Changing a Remote Node Entry.....	5-49
Removing Remote Nodes.....	5-51
Procedure for Removing a Remote Node.....	5-51
Example: Removing a Remote Node.....	5-51
Example Notes: Removing a Remote Node.....	5-51
Summary.....	5-52
Concepts.....	5-52
Customizing System Startup Files.....	5-52
Customizing Login Command Procedures.....	5-52
Customizing System Parameters.....	5-53
Maintaining the DECnet for OpenVMS Remote Node Database.....	5-53
Commands.....	5-54
Customizing System Startup Files.....	5-54
Customizing Login Command Procedures.....	5-54
Customizing System Parameters.....	5-54
Maintaining the DECnet for OpenVMS Remote Node Database.....	5-55

Module 6 — Installing Layered Products

About This Module.....	6-2
Introduction.....	6-2
Objectives.....	6-3
Managing Product Licenses.....	6-4
Overview.....	6-4
Topics.....	6-4

The License Management Facility.....	6-5
Components of the LMF	6-5
License Units.....	6-6
Activity Licenses	6-7
Product Usage with Activity License in a Cluster	6-8
License Management Utility	6-9
Entering LICENSE Commands.....	6-9
Privileges Needed to Use LICENSE	6-10
Listing Licenses.....	6-10
Loading and Unloading Licenses	6-10
Enabling Licenses.....	6-10
Sample Product Authorization Key.....	6-11
Example: VMSLICENSE Session.....	6-12
License Management Facility and License Agreements	6-14
Hints on Running VMSLICENSE.....	6-14
Installing Layered Products.....	6-15
Overview	6-15
Topics.....	6-15
Adjusting User Privileges and Quotas.....	6-16
Privileges and Quotas	6-16
How to Adjust User Privileges and Quotas	6-16
DEFAULT Account	6-16
Quotas and Resource Limits	6-17
AUTHORIZE Qualifiers for Quota Fields.....	6-17
Adjusting System Parameters.....	6-18
Ensure Number of GBLPAGES and GBLSECTIONS	6-18
Example: Output from the LIST COMMAND	6-18
Example: SYSGEN SHOW Command	6-19
Example Notes: SYSGEN SHOW Command	6-19
Commonly Changed System Parameters.....	6-20
Installing Layered Products on a Common System Disk	6-21
Things to Know Before Installing a Product	6-21
Considerations When Installing Products on an OpenVMS	
Cluster System.....	6-21
When to Run AUTOGEN	6-22
After Installation.....	6-22
Summary of the VMSINSTAL Options	6-23
Installations Using the POLYCENTER (PCSI) Software Installation Utility...	6-24
POLYCENTER Software Installation Utility.....	6-24
Privilege Needed for the POLYCENTER Software Installation Utility ...	6-25
Using the DECwindows Motif Interface	6-25
Using the DCL Keyboard Interface.....	6-26
Summary	6-27
Concepts.....	6-27
Managing Product Licenses	6-27
Installing Layered Products	6-28

Commands.....	6-29
Managing Product Licenses	6-29
Installing Layered Products	6-29

Module 7 — Monitoring System Resources

About This Module.....	7-2
Introduction.....	7-2
Objectives.....	7-2
Using the Show Cluster Utility.....	7-3
Overview	7-3
Topics.....	7-3
Show Cluster Utility	7-4
Running SHOW CLUSTER	7-4
Example: SHOW CLUSTER Default Display.....	7-5
Example Notes: SHOW CLUSTER Default Display.....	7-5
Additional SHOW CLUSTER Display Fields	7-5
Tailoring the SHOW CLUSTER Display.....	7-6
Adding and Removing Fields.....	7-6
Example: Adding and Removing Fields	7-6
Example Notes: Adding and Removing Fields	7-6
Changing the Width of Fields	7-7
SHOW CLUSTER Command Summary.....	7-7
Using a SHOW CLUSTER Initialization File	7-8
SHOW CLUSTER Initialization Files.....	7-8
Creating and Using an Initialization File	7-8
Example: SHOW CLUSTER Initialization File	7-9
Monitoring with DCL Commands.....	7-10
Overview	7-10
Topics.....	7-10
Monitoring Memory Resources.....	7-11
Memory Resources	7-11
Page Size in Memory	7-11
SHOW MEMORY Command.....	7-11
Example: SHOW MEMORY Command on an OpenVMS VAX System	7-12
Example Notes: SHOW MEMORY Command on an OpenVMS VAX System.....	7-12
Example: SHOW MEMORY Command on an OpenVMS Alpha System	7-14
Example Notes: SHOW MEMORY Command on an OpenVMS Alpha System.....	7-14
Using SHOW SYSTEM in a Cluster.....	7-15
SHOW SYSTEM Command.....	7-15
Qualifiers Useful in a Cluster.....	7-15
Example: SHOW SYSTEM /CLUSTER on an OpenVMS System	7-16
Example: SHOW SYSTEM/NODE on an OpenVMS VAX System.....	7-17

Monitoring the Activity of a Process.....	7-18
Monitoring Process Activity	7-18
SHOW PROCESS /CONTINUOUS Command	7-18
Privileges Required.....	7-19
Example: SHOW PROCESS /CONTINUOUS	7-19
Example Notes: SHOW PROCESS /CONTINUOUS	7-19
Using the Monitor Utility.....	7-20
Overview	7-20
Topics.....	7-20
Monitor Utility.....	7-21
Running the Monitor Utility.....	7-21
Monitor Commands	7-21
Interrupting the MONITOR Display	7-21
Example: MONITOR SYSTEM Display	7-22
Example Notes: MONITOR SYSTEM Display.....	7-22
Adjusting the Sampling Interval.....	7-23
Adjusting the Display Interval	7-23
Creating a Recording File	7-23
Creating a Summary File	7-23
Monitoring Processes.....	7-24
PROCESSES Class.....	7-24
Specifying Resources to Monitor	7-24
Example: MONITOR PROCESSES /TOPCPU	7-25
Example Notes: MONITOR PROCESSES /TOPCPU	7-25
Using MONITOR in an OpenVMS Cluster System.....	7-26
CLUSTER Class.....	7-26
Example: MONITOR CLUSTER	7-27
Other Qualifiers Useful in a Cluster	7-28
Example: MONITOR /NODE.....	7-28
Multinode Summary	7-29
Procedure to Record Data for a Multinode Summary	7-29
Procedure to Start Recording on Two Nodes.....	7-30
Example: Creating a Multinode Summary.....	7-31
MONITOR Command Procedures	7-32
Command Procedures	7-32
SUBMON.COM.....	7-33
Example: SUBMON.COM	7-33
MONITOR.COM	7-34
Example: MONITOR.COM.....	7-34
MONSUM.COM.....	7-36
Monitoring the Network with NCP Commands.....	7-38
Overview	7-38
Topics.....	7-38
Monitoring DECnet Lines.....	7-39
DECnet Lines	7-39
Identifying Lines	7-39
Displaying Lines.....	7-40

Monitoring DECnet Circuits	7-41
DECnet Circuits.....	7-41
Identifying Circuits.....	7-41
Circuit Identifier	7-41
Displaying Circuits	7-42
Example: Displaying Circuits	7-42
Circuit States	7-42
Monitoring DECnet Links.....	7-43
DECnet Links	7-43
Displaying Links	7-43
Example: Displaying Links.....	7-44
Example Notes: Displaying Links.....	7-44
Summary	7-45
Concepts.....	7-45
Using the Show Cluster Utility	7-45
Monitoring with DCL Commands.....	7-45
Monitoring the Network with NCP Commands.....	7-46
Commands.....	7-47
Using the Show Cluster Utility	7-47
Monitoring with DCL Commands.....	7-47
Monitoring the Network with NCP Commands.....	7-47

Module 8 — Understanding Security Concepts

About This Module.....	8-2
Introduction.....	8-2
Topics.....	8-2
Objectives.....	8-2
General System Security.....	8-3
Overview	8-3
Topics.....	8-3
Physical Security Consideration.....	8-4
Physical Security	8-4
Levels of Required Security.....	8-4
Examples of Physical Security	8-5
Software Security	8-6
Minimum Software Security	8-6
Additional Security.....	8-6
OpenVMS Cluster Security Considerations.....	8-7
Security Management in an OpenVMS Cluster System.....	8-7
Building a Common Environment.....	8-7
Required System Files	8-7
System Files That Must Be Common in a Cluster	8-8
User Security Profile.....	8-9
Overview	8-9
Topics.....	8-9

User Identification Code	8-10
UIC Group Identifier	8-10
UIC Protection.....	8-10
Assigning UICs	8-10
Suggestions for Dividing Users into Groups	8-11
Process Interaction.....	8-12
Effect of Group UIC Assignment.....	8-12
Rights Identifiers	8-13
Rights Identifiers	8-13
Types of Identifiers.....	8-13
Environmental Identifiers	8-13
General Identifiers	8-13
UIC Identifiers.....	8-14
Facility Identifiers.....	8-14
System Node Identifiers.....	8-14
Example: Displaying Rights Identifiers.....	8-15
Example Notes: Displaying Rights Identifiers.....	8-15
Privileges.....	8-16
OpenVMS Privileges	8-16
Minimum Security Privileges.....	8-16
Types of Privileges	8-16
Assigning Privileges	8-17
Categories of Privileges	8-18
Possible Privilege Problems.....	8-20
Restricting User Activities	8-21
Overview	8-21
Topics.....	8-21
Tailoring User Accounts	8-22
Example: UAF Field Categories	8-23
Example Notes: UAF Field Categories	8-23
Access and Security Fields.....	8-24
Access Times and Modes.....	8-24
Restricting Access by Time of Day.....	8-24
Restricting Access by Day of Week	8-24
Restricting Access by Access Mode.....	8-24
Login Access Modes.....	8-25
AUTHORIZE Qualifiers for Access Fields	8-25
Security Fields.....	8-26
Login Flags.....	8-27
Login Flag Parameters	8-27
Creating and Using Captive Accounts.....	8-28
Captive Accounts.....	8-28
Characteristics of Captive Accounts.....	8-28
Steps for Setting Up a Captive Account	8-29
Example: Setting Up a Captive Account	8-30
Example: Login to a Captive Account.....	8-30

Example Notes: Login to a Captive Account.....	8-30
Example: Captive Command Procedure.....	8-31
RESTRICTED and CAPTIVE AUTHORIZE Flags.....	8-31
Object Security Profile.....	8-32
Overview	8-32
Topics.....	8-32
Ownership and Protection of Objects	8-33
Ownership Element	8-33
Who Can Change Ownership?	8-33
Source of a File's Owner Identifier.....	8-33
Protection	8-34
Access Control Lists	8-35
Why Access Control Lists Are Needed	8-35
Access Control Lists	8-35
Determining if an Object Has an ACL	8-35
Displaying an ACL	8-35
How Access Is Determined.....	8-36
Figure: Object Access	8-36
Step 1: ACL Check.....	8-37
Step 2: Protection Code Check.....	8-37
Step 3: Privilege Check.....	8-38
Step 4: Access Overrides	8-38
Manipulating an Object's Security Profile	8-39
SET SECURITY Command	8-39
Displaying an Object's Security Profile	8-39
Example: Displaying an Object's Security Profile	8-39
Example Notes: Displaying an Object's Security Profile	8-39
Object Classes	8-40
Example: Changing the Security Profile of a Nonfile Object.....	8-41
Example Notes: Changing the Security Profile of a Nonfile Object.....	8-41
Required Access for Changing an Object's Security Profile.....	8-41
Example: Modifying an Object's Security Profile.....	8-42
Example Notes: Modifying an Object's Security Profile.....	8-42
Using Passwords to Maintain System Security.....	8-43
Overview	8-43
Topics.....	8-43
Defining User Passwords.....	8-44
Password Options	8-44
Requiring User Passwords	8-44
Example: Setting User Passwords.....	8-44
Requiring a Secondary Password	8-45
Requiring a Generated Password	8-45
Filtering Passwords.....	8-46
Dictionary Search	8-46
History Search	8-46

Defining System Passwords	8-47
System Password	8-47
Defining a System Password	8-47
Steps for Defining a System Password	8-47
Changing System Passwords	8-47
Examples of Password Usage	8-48
Example: Normal Password Login	8-48
Example: Secondary Password Login	8-48
Example: System Password Login	8-48
Summary	8-49
Concepts	8-49
General System Security	8-49
User Security Profile	8-50
Restricting User Activities	8-50
Object Security Profile	8-51
Using Passwords to Maintain System Security	8-51
Commands	8-52
User Security Profile	8-52
Restricting User Activities	8-52
Object Security Profile	8-52
Using Passwords to Maintain System Security	8-53

Glossary

Learning Check

Learning Check Answers

About This Course

Introduction

This About This Course section describes the contents of the course and suggests ways to use the material most effectively.

Topics

The following topics are discussed:

- Course description
- Prerequisites
- Resources
- Course organization
- Course goals
- Course nongoals
- Course conventions

Course Description

OpenVMS System and Network Node Management II continues the system management training presented in *OpenVMS System and Network Node Management I*. This course prepares the student to modify an established system that is part of a clustered and/or networked environment.

This course provides students with the information and guided practice they need to perform the tasks required of system managers in an operational OpenVMS VAX, OpenVMS Alpha, or VMScluster system environment. These tasks include managing disks and queues, installing layered products, monitoring system resources, and performing basic system customization tasks. Further, the course enables the system manager to manage LAT services, such as terminal services and remote print queues.

Prerequisites

Students taking this course should be able to:

- Manage system users, which requires:
 - Maintaining the user authorization file (UAF) and volume quota files
 - Creating user file directories (UFDs)
 - Controlling user processes
- Manage system resources, which requires:
 - Mounting and dismounting disk and tape volumes
 - Setting device characteristics
 - Starting and stopping print and batch queues
 - Performing full and incremental backups and restores of disk volumes
 - Starting up and shutting down a system using the default startup procedure

These requirements can be met by completing the *OpenVMS System and Network Node Management I* course.

Resources

Students should have access to the following manuals to perform the recommended learning activities of this course.

- *DECams User's Guide*
- *Digital OpenVMS System Messages: Companion Guide for Help Message Users*
- *Guide to the DEC Text Processing Utility*
- *Guidelines for OpenVMS Cluster Configurations*
- *OpenVMS Alpha Version 7.2 Upgrade and Installation Manual*
- *OpenVMS Alpha Galaxy Guide*
- *OpenVMS Guide to System Security*
- *OpenVMS Cluster Systems*
- *OpenVMS DCL Dictionary*
- *OpenVMS License Management Utility Manual*
- *OpenVMS Management Station Overview and Release Notes*
- *OpenVMS System Management Utilities Reference Manual*
- *OpenVMS System Manager's Manual*
- *OpenVMS User's Manual*
- *OpenVMS VAX Version 7.2 Upgrade and Installation Manual*
- *OpenVMS Version 7.2 New Features Manual*
- *OpenVMS Version 7.2 Release Notes*
- *Overview of OpenVMS Documentation*
- *TCP/IP Networking on OpenVMS Systems*
- *VAX MACRO and Instruction Set Reference Manual S Volume Shadowing for OpenVMS*

Course Organization

This course is organized into a series of modules. Each module has its own learning objectives and covers a single topic or group of closely related topics. A module consists of:

- An **introduction** describing the purpose of the module.
- One or more **objectives**, which identify the skills taught in the module. Objectives are designed to focus the student's study efforts on a selected number of skills.
- The module **resource list**, which lists individual reference documents that support the information in the module.
- The module **text** that consists of:
 - Descriptive text organized by topic
 - Illustrations that clarify the relationships among various elements of an OpenVMS system, or summarize steps of a particular process or command
 - Examples containing sample listings from actual interactive sessions on an OpenVMS system
- A module **summary** that reviews important concepts and commands taught in the module.

Written and laboratory **exercises** are also provided with this course. These exercises help students to review and practice the skills learned during the lecture session.

The **glossary** contains definitions for many terms used in this course.

Course Goals

To modify an established system, a system and network node manager should be able to:

- Describe major system resources and their function in an OpenVMS environment.
- List the components of a DECnet network.
- Create and use queues, including forms definition and management.
- Define and manage LAT services.
- Manage disk and tape volumes.
- Describe the functions of the system startup and login files.
- Set system parameters using AUTOGEN and SYSMAN.
- Install OpenVMS layered products and updates.
- Monitor system resources.
- Perform tasks that enhance system security.

Course Nongoals

This course does not cover the following topics:

- Installation of the OpenVMS operating system.
- System performance management and tuning.
- System troubleshooting.
- Programming using system services, run-time libraries, or other OpenVMS features.
- Advanced OpenVMS concepts or system programming.
- Details of layered product features and functions.

Course Conventions

This table describes the conventions used in the listings and command tables of this course.

UPPERCASE \$ TYPE OPERATOR.LOG	In commands, uppercase characters indicate words that you type exactly as they appear. For example, you would type the command on the left as it appears.
Lowercase \$ TYPE file-spec	Lowercase characters represent elements that you must replace according to the description in the text. For example, you must follow certain rules when you replace "file-spec" in the command on the left.
Horizontal ellipses \$ TYPE file-spec,...	Horizontal ellipses indicate that you can enter additional parameters, values, or information. For example, you can enter any number of file specifications in this TYPE command.
Vertical ellipses \$ TYPE MYFILE.DAT . . .	Vertical series of periods or ellipses mean that not all of the data that the system would display in response to the particular command is shown, or that not all of the data a user would enter is shown.
Square brackets ([]) \$ SHOW PROCESS [process-name]	Square brackets indicate that the enclosed item is optional in the context of a syntax statement. For example, the process name is optional in the command shown. (Square brackets are not optional, however, when specifying directories in the syntax of a file specification.)

Materials

Introduction

This instructor guide has two sections. The first has general information about the course including teaching strategies, suggested schedules, and other hints. The second consists of the student materials, with additional detail and teaching notes.

Read the student information at the beginning of this About This Course section. It lists the goals and prerequisites for the course.

This course consists of alternate lectures and labs and is designed to be taught in one week. Students receive a course guide, including both lecture and lab materials.

The instruction is divided into units called **modules**. Each module covers a skill or related group of skills required to fulfill the course goals.

Student Package

Each student in the class receives a student package containing:

- Course Guide
- Exercises

Instructor Package

Each instructor should receive an Instructor Package containing:

- This Instructor Guide

In addition, provide a full OpenVMS document set in the lab or classroom for reference.

The course guide is designed as an aid for the student during lecture. As you explain a particular subject, you can refer the student to the corresponding outline in the guide, and the student can take notes in the blank areas provided. By using the topic outlines, tables, figures, and examples as a framework, the student will be able to channel his or her energies into understanding the subject matter and away from recording it.

Transparencies

Many instructors like to illustrate their lectures with transparencies duplicated from the pages of the course guide. To keep this textbook to a manageable size, the lettering in the outlines is not large, and the lettering in the tables and examples is smaller still. Your transparencies will be more effective if you have them typed using large lettering.

Using Tables

Many of the system manager's functions are explained using tables. Most tables are organized according to the function to be performed, not the utility or command used. It is much more interesting to learn how to perform a sequence of useful jobs than it is to study an alphabetical sequence of commands and qualifiers.

The tables usually contain command formats, examples, and explanatory comments. They are intended for use in the lab and on the job as well as in class. When you lecture, you are not expected to teach every function in every table, nor to restrict yourself to what is in the tables. By all means, explain the material in your own way. But refer to the tables as you go along, so that the student will be familiar with them when he or she is in the lab and on the job.

Hardware Needed

Many instructors find it useful to have their students try out commands as they are taught, and we encourage you to do this as well. To do this there should be at least one terminal available for every two students.

Preparing to Teach

Introduction

To prepare to teach this course, you need the following:

- An instructor package
- A copy of each document listed under "Additional References"
- The OpenVMS documentation set

Preparing the System

Before offering this course, you must prepare a system to support the laboratory exercises that accompany the written materials. To prepare for the course, complete the following steps:

- Select a system.
- Create student accounts.

Selecting Equipment

The following equipment must be available to conduct the course. See the VMS Software Product Description (SPD) for further information about supported configurations and memory requirements.

- An Alpha system running Version 7.2 or a higher version of OpenVMS
- A terminal for each student

The exercises for this course call for installing, starting up, and reconfiguring the OpenVMS system in many ways. These activities interfere with students using the system for other classes. Furthermore, students with the privileges necessary to perform these exercises may unintentionally affect system behavior in many undesirable ways.

For these reasons, it is strongly recommended that the labs be run on a standalone machine with disk volumes dedicated exclusively to courses in which students have unlimited privileges.

Creating Student Accounts

You should create a separate account for each student who enrolls in a course.

Each student should also be allocated a disk quota of at least 1000 blocks if quotas are enabled. None of the operations performed by students require enhanced quotas or other special parameter values. Allow these values to default to the values specified for your training center.

The student accounts should have the following privileges, at minimum:

- TMPMBX – allows use of queues
- NETMBX – allows use of DECnet software

Other privileges will be needed to perform some operations described in the course. Those privileges are potentially dangerous, and you may want to restrict their use. Some operations also require physical access to the system hardware and must be done under your supervision.

If the course is given on a standalone system, allow each student to have the SETPRV privilege and allow students to give themselves the privileges they need to perform various operations.

Understanding System Resources

Module 1

About This Module

Introduction

The role of the operating system is to share resources among all processes running on the system. Memory is an important resource and while it is common to think of memory only in terms of how much data can be stored, the speed at which the CPU accesses the data is important to the performance of the computer.

Other resources that must be shared include CPU time and mass storage.

Objectives

To demonstrate an understanding of OpenVMS system resources, you should be able to:

- List the main features of virtual memory management in an OpenVMS system.
- List the characteristics of a working set.
- Define the following terms: paging, swapping.
- Describe how an OpenVMS system schedules resources.
- Identify common network configurations.

Virtual Memory Management Concepts

Overview

The OpenVMS operating system is a multiuser, virtual memory management system. The operating system is responsible for scheduling the use of system resources such as memory and CPU time.

Topics

In this section the following topics are discussed:

- Virtual and physical memory
- Virtual address space layout

Virtual and Physical Memory

A **virtual memory** management system allows the execution of programs or images that may not reside completely in physical memory. A large program can run on a computer with small memory capacity.

Each process has a share of physical memory. At any one time, only a part of the program is in physical memory. The part of the program not in memory is stored on disk.

The operating system is responsible for sharing memory among programs. Memory management is invisible to the programmer.



Note

The term physical memory refers to internal memory, excluding cache and registers. A user program that is larger than the physical memory of the computer has part of the program in physical memory and part stored on disk. To the user, the entire program appears to be in memory.

The user's view of memory is that of one large, contiguous space starting at address 0. This does not exist in hardware; therefore, the memory management system is referred to as **virtual**.

Virtual Address Space

For the virtual address space under OpenVMS:

- Each byte of memory is individually addressable.
- The first byte is located at virtual address 0.
- Virtual memory is divided in the following ways:
- On a VAX system, virtual memory is divided into **pages** of 512 bytes.
- On an Alpha system, a 512-byte quantity is called a **pagelet**, while a page size will be one of four values: 8 kilobytes (KB) (8192 bytes), 16 KB, 32 KB, or 64 KB.

A particular Alpha system will implement only one of the four pages sizes; the initial set of Alpha computers use an 8 KB page. Internally, for the purposes of memory allocation, deletion, and protection, OpenVMS Alpha rounds up (if necessary) the value supplied in pagelets to a number of CPU-specific pages.

- Virtual pages are numbered starting at VPN 0 (virtual page number).

Physical Address Space

The amount of physical memory varies from one implementation of the architecture to another.

- Physical memory is divided in the following ways:
 - On a VAX system, physical memory is divided into 512 byte pages.
 - On an Alpha system, physical memory is divided into pages whose size can be 8 KB, 16 KB, 32 KB, or 64 KB.
- Physical pages are numbered starting at PFN 0 (page frame number).
- A program can have more virtual pages than the system has physical pages.
- Each process has a portion of its virtual pages in physical memory.
- Virtual pages that are not in physical memory are stored on disk.

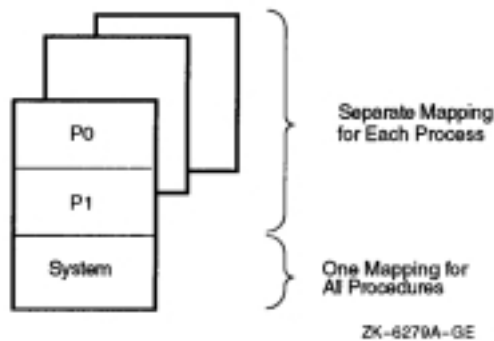
Virtual Address Space Layout

The virtual address space is divided into:

- System Space
 - System services and RMS
 - OpenVMS VMS code and data
 - Paged and nonpaged pool
- Process space for each process
 - P0 space—image, run-time libraries, debugger
 - P1 space—process-specific control information, command language interpreter

Layout of Virtual Address Space

The following figure shows how virtual address space is mapped:



Note

The operating system is located in system space and is shared by all processes running on the system. User programs call operating system routines to perform services.

Other data structures contained in system space include:

- Tables that contain information on each process running on the system
- System and group logical name tables
- Mailboxes

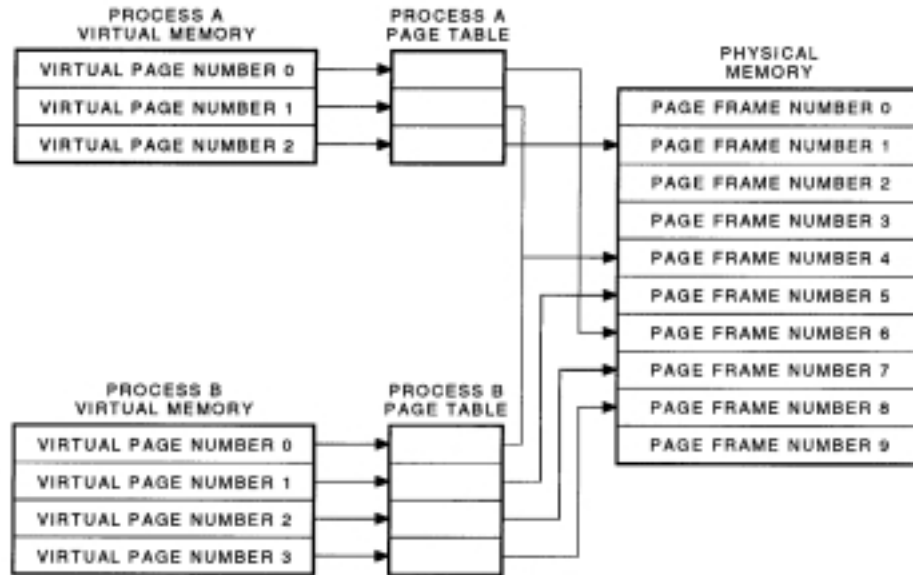
Every process has its own unique process space, which is divided into two regions.

- Program Region or P0 space is reserved for the current image or program code that is executing.
- Control Region or P1 space is used for the four per-process stacks and other control functions.

A process cannot typically access the P0 or P1 space of any other process.

Virtual Address Space Mapped to Physical Memory

Each process has a page table that maps virtual pages to physical pages. The physical pages are in physical memory or on disk. The page frame number is the address of the physical page in memory. The following figure shows how virtual address space is mapped to physical memory:



TTB_X0595_BBA



Note

The system parameter VIRTUALPAGECNT defines the maximum size of each process page table (including both P0 and P1 entries).

- Must be large enough for the largest application.
- Installation guides for most layered products recommend minimum value of VIRTUALPAGECNT.
- User message "Insufficient virtual memory" may indicate VIRTUALPAGECNT is too small.

There is only one page table for system space. The hardware uses this page table to translate virtual addresses located in system space to physical addresses.

Working Sets

Overview

Memory resources from a process point of view are defined by the **working set** assigned to that process.

Topics

In this section the following topics are discussed:

- Working set concepts
- Working set parameters

Working Set Concepts

At any one time, some of the pages of a process' virtual address space are located on disk and some in physical memory.

- Each process has a limit on the number of physical pages of memory that can be used to map their virtual address space.
- The set of physical pages currently being used by a process is called the **working set**.
- Virtual pages not in the working set are either:
 - On disk
 - In physical memory on the free page list or the modified page list

Working Set Limit

As the program executes, the number of virtual pages occupying physical pages in the working set grows and approaches the working set limit.

When a process requests another page beyond this limit, one page currently in the working set must be released to accommodate the new page.

Under certain conditions the working set *is* allowed to grow.

Parameters Controlling Working Set

Three values are associated with the size of a working set:

- WSDEFAULT is the initial limit to the size of a process working set.
- WSQUOTA is the size to which the working set is allowed to grow if the process is page faulting heavily.
- WSEXTENT is the maximum to which the working set is allowed to grow if the process is page faulting heavily and there is sufficient free memory on the system.



Note

Beginning with OpenVMS Version 7.2, the system will assign a new interactive process the larger of WSEXTENT or the SYSGEN parameter PQL_MWSEXTENT. Unless you assign another value dynamically or in MODPARAMS.DAT, PQL_MWSEXTENT is set to WSMAX. Thus, a process may be allowed to use more memory than specified by the settings in AUTHORIZE

Working Set Parameters

An interactive process obtains working set values from the user authorization file (UAF). The system manager sets these parameters when the account is created.

Example: Working Set Parameters for Interactive Processes

Use the Authorize utility to display and change working set parameters for users as in the following example:

```
$ SET DEFAULT SYS$SYSTEM:
$ RUN AUTHORIZE
UAF> SHOW USER1
Username: USER1      Owner: SAMPLE USER
Account: VMS          UIC: [300,3] ([VMS,USER1])
CLI: DCL              Tables: DCLTABLES
Default: USER1:[USER1]
LGICMD: LOGIN
Flags:
Primary days: Mon Tue Wed Thu Fri
Secondary days:      Sat Sun
No access restrictions
Expiration: (none) Pwdminimum: 15 Login Fails: 0
Pwdlifetime: 90 00:00 Pwdchange: 18-OCT-1999 16:21
Last Login: 16-NOV-1999 13:08 (interactive), 1-NOV-1999 18:24 (non-
interactive)
Maxjobs: 0 Fillm: 40 Bytln: 32768
Maxacctjobs: 0 Shrfillm: 0 Pbytln: 0
Maxdetach: 0 BIOltn: 18 JTquota: 1024
Prclm: 10 DIOltn: 18 W$def: 256
Prio: 4 ASTltn: 24 W$quo: 512
Queprio: 0 TQEln: 20 W$sextent: 2048
CPU: (none) Enqltn: 200 Pgflquo: 40000
Authorized Privileges:
NETMBX SETPRV TMPMBX
Default Privileges:
NETMBX SETPRV TMPMBX
```

Example Notes: Working Set Parameters for Interactive Processes

Notes on the Working Set Parameters for Interactive Processes example include:

- ❶ WSDEFAULT, working set default
- ❷ WSQUOTA, working set quota
- ❸ WSEXTENT, working set extent



Note

The point here is to show that the parameters for interactive processes are set by the user authorization file. Avoid going into detail about these or other parameters and the limits shown on this screen.

Students can also use the DCL command SHOW WORKING_SET to display the parameters set for their interactive process. The display of the account profile is used to emphasize that these parameters are controlled by the system manager at account setup time.

```
$ SHOW WORKING_SET
Working Set /Limit= 1024 /Quota= 6144 /Extent= 10240
Adjustment enabled Authorized Quota= 6144 Authorized
Extent= 10240
```

Working Set Parameters for Batch Processors

Working set values for batch processes are taken from:

- The queue itself (if established by the system manager)
- The SUBMIT command using various qualifiers
- The account profile values of the process that submitted the job



Note

This discussion is to introduce the idea of setting different working set limits for batch processes.

Example: Displaying Queue Parameters

Use the SHOW QUEUE command to display batch queue characteristics.

```
$ SHOW QUEUE/BATCH/FULL
Batch queue BARNUM_BATCH, idle, on BARNUM::
  /AUTOSTART_ON=(BARNUM::) /BASE_PRIORITY=3 /NOENABLE_GENERIC /JOB_LIMIT=4
  /OWNER=[SYSTEM] /PROTECTION=(S:M,O:D,G:R,W:S)
  /WSDEFAULT=10000 /WSEXTENT=30000 /WSQUOTA=20000

Generic batch queue DOC$BATCH
  /GENERIC=(BARNUM_BATCH,DOGS_BATCH) /OWNER=[SYSTEM]
  /PROTECTION=(S:M,O:D,G:R,W:S)
.
.
.
Batch queue STUDENT$BATCH, stopped, on BARNUM::
  /BASE_PRIORITY=4 /JOB_LIMIT=4 /OWNER=[SYSTEM] /PROTECTION=(S:M,O:D,G:R,W:S)
```



Note

Use this example to show that different queues can be set up for different types of jobs. Set the stage for the following chapter by mentioning the base priority setting for the queues. Also discuss that queue protection can be used to restrict which jobs are submitted to which queues.

Working Set Parameters for Other Processes

Working set values for other processes can be controlled by:

- Passing the values as arguments if the processes were created using the \$CREPRC system service
- Passing the values on the DCL RUN/DETACH command

Default Working Set Values

If working set values are not supplied when a process is created, the associated system parameter values will be used.

The following table lists parameters that establish working set values.

To establish this value...	Set this parameter:
The minimum WSDEFAULT for any process.	PQL_MWSDEFAULT
The default value for WSDEFAULT.	PQL_DWSDEFAULT
The minimum WSQUOTA for any process.	PQL_MWSQUOTA
The default value for WSQUOTA. The value of WSQUOTA cannot exceed the value of WSMAX.	PQL_DWSQUOTA
The minimum WSEXTENT for any process.	PQL_MWSEXTENT
The default value for WSEXTENT.	PQL_DWSEXTENT
The maximum working set size for any process. The highest value WSMAX can have is 65,535 pages.	WSMAX

Paging and Swapping

Overview

Two methods used by the OpenVMS system to manage memory resources are paging and swapping. Both methods use one or more disk devices as an extension of physical memory.

Topics

In this section the following topics are discussed:

- Paging
- Swapping

Paging

Paging is the action of bringing pages into the working set of a process.

A **page fault** occurs when a process refers to a page that is not in its working set.

Page Faults

When a page fault occurs:

- The valid bit in the page table entry is clear.
- The hardware generates a translation-not-valid exception.
- The system runs the page fault handler (pager) to move the needed page into the working set.



Note

Be aware that the term **working set** may have a different meaning outside of the OpenVMS documentation set. Other sources define working set as the set of pages that has been referenced in a certain time interval. When this definition of working set is used, the set of pages in physical memory is referred to as the resident set.

This course uses the term working set as it is used in the OpenVMS documentation. The working set is the set of pages in physical memory in the process-specific space, or the set of pages for which the valid bit is set in the page table entry.

A virtual page may occupy:

- A physical page listed in the working set
- A physical page listed in the free or modified page list
- No physical page

A virtual page that has been referenced recently may be on disk. The use of the free and modified page lists as a cache for pages recently replaced from the working set helps to lower the probability that a recently referenced page is not in physical memory.

Image Startup

For a process to execute an instruction that is part of an image, that instruction must be in memory. The following sequence of events occurs with image startup.

1. A system routine is invoked to create the page table to map the virtual addresses to the physical addresses.
2. The initial page table entries indicate that the pages of the image are stored on disk.
3. Control is transferred to the image.
4. When the CPU attempts to access the first instruction, the address translation hardware finds that the page of the image is not in memory.
5. A page fault occurs.



Note

This is an example of demand paging. A virtual page is not brought into physical memory until it is referenced. Students with a background in computer science should realize this to be true at this point. Students without such a background will not care.

The opposite of demand paging is prepaging. In a prepaging operating system a number of virtual pages are read into physical memory when the image is started. The OpenVMS system actually does some prepaging in that pages are read into memory in clusters.

Always specify virtual page and physical page and ensure that the students understand what you mean.

Discuss how WSDEFAULT sets the initial working set size limit for a process.

Page Table

When a program references a virtual page, the hardware translates the virtual page to a physical page.

The page table is the data structure used to map virtual pages to physical pages.

- There is a page table for each process.
- The page table contains an entry for each page in virtual memory.

Page Table Entry

The page table entry locates the contents of the virtual page in:

- Physical memory and currently in use by the process
- Physical memory and on the free or modified page list
- A file on disk

If the page is currently in use by the process, the page table entry contains a:

- Flag that indicates the page is in the process working set (valid bit)
- Pointer to the page in physical memory
- Flag that indicates whether the page has been changed since it was read into physical memory from the disk (modify bit)
- Protection code

If the page is not in physical memory, the page table entry contains a pointer to the location of the page on disk.

Global Pages

Pages that are shared among multiple processes are called **global pages**.

- There is one physical copy of a global page.
- Each process has a page table entry for the global page.
- The global page is counted toward the working set size of each process that maps it.

Page Fault Types

There are two types of page faults.

- A **hard fault** requires disk I/O to obtain the virtual page from the:
 - Original disk
 - Page file
- A **soft fault** is satisfied without disk I/O. The virtual page is in memory but outside the working set, located on the:
 - Free page list
 - Modified page list



Note

A soft fault can be a demand-zero fault. This occurs when a program allocates spaces for variables. There is no page on disk to read in because the program is just asking for space in which to store data that will be generated in the future.

A soft fault can also be to a global page.

The statement that a hard fault always occurs when a virtual page is accessed for the first time is not necessarily true if the page is shared and happens to be in memory as the result of an access by another process. However, to keep the discussion simple, we ignore this possibility.

Hard Faults

When a hard page fault occurs, the operating system has to:

- Locate the virtual page in the image file on disk.
- Allocate a physical page of memory and store the contents from disk.
- Update the page table entry for the virtual page to reflect the new location of the page.
- Add the page to the appropriate working set.

Removing a Page from the Working Set

When a process whose working set has reached the limit tries to access an additional page, a page must be removed from the working set to make room for the new page.

- A page in the working set is selected for replacement.
- If this page has not been modified, it is placed on the bottom of the free page list.
- If this page has been modified, it is placed on the bottom of the modified page list.
- The new page is retrieved and now added to the working set.

Free Page List

The **free page list** is a list of pages that are available to the process for allocation.

- When a process generates a hard page fault, the page at the head of the free page list is appropriated for that process.
- The list is maintained in the order in which the pages arrive.
- Over time, a page that is not referenced again will work its way to the head of the free page list.
- A page on the free page list that is referenced again is returned to the working set of the appropriate process (soft page fault).
- When a page is removed from the free page list, the page table for the process losing the page is changed to reflect that the page is no longer in physical memory.

**Note**

Both the free page list and the modified page list are global and are used by all processes.

The free page list and the modified page list serve as a cache for the disk. Because of locality of reference, a page that has been recently referenced is likely to be referenced again. Because of the free and modified page list, the page is likely to still be in memory.

Students with a background in computer science may have questions about the algorithm for selecting a page for replacement. The OpenVMS operating system uses a modified first-in/first-out (FIFO) replacement algorithm.

The FIFO selection algorithm is not optimal. If any student points this out you might want to discuss that:

- The free and modified page lists provide a cache of recently used pages in memory.
- The translation buffer cache tends to keep recently used pages in the working set.

Because of the above modifications of the FIFO algorithm, virtual memory works better than just a plain FIFO policy. In fact, virtual memory works very well on the VAX.

Page replacement policies that try to keep track of when a page was last used employ a "use bit." There is also processing overhead to these policies. The VAX does not have a "use bit."

Modified Page List

The **modified page list** is a list of pages that have been modified since being read from disk.

- This list is maintained in the order in which the pages arrive.
- When the modified page list reaches a certain size:
 - The modified pages are written to a special system file called a **page file**.
 - A process pointer is updated to reflect the new location of the virtual page.
 - The pages are placed on the free page list.
- Pages on the modified page list can be returned to the working set if they are referenced again.



Note

Why are there two lists? Why treat a modified page differently from an unmodified page? The reason is to save overhead.

- A modified page can be recalled and changed again, for example, as in the counter to a program loop.
- Unmodified pages do not have to be written to disk; they are already there.

The process pointer that is updated is actually the BAK array in the PFN database, which will be used at a later time to update the PTE if the page should get allocated by another process.

Page Fault Read Clustering

Pages can be read into memory from disk in groups called **page fault clusters**. Maximum page fault cluster sizes are set by:

- System parameter PFCDEFAULT
- Linker option CLUSTER, which overrides PFCDEFAULT

During the execution of programs, PFCDEFAULT controls the number of image pages read from disk per I/O operation when a page fault occurs. The read I/O operations can take place from an image file or from the page file. The actual size of the cluster can be less than PFCDEFAULT, depending on the size of image sections and the pattern of page references.

PFCDEFAULT can be overridden on an image-by-image basis with the CLUSTER option of the OpenVMS Linker.

Swapping

Swapping refers to writing a process working set out to a reserved disk file known as the swap file. This allows the remaining processes to use more memory.

The goal of swapping is to free up memory being used by idle processes and to make it available for the active processes.

SYS\$SYSTEM:SWAPFILE.SYS is the file used for swapping. The system page file can also be used for swapping.

SWAPPER Process

The SWAPPER process is an integral part of the OpenVMS operating system. The functions of the SWAPPER include:

- Inswapping the SHELL, if necessary, to serve as a template for process creation
- Maintaining the size of the modified page list
- Maintaining the number of free pages available for processes
- Swapping process working sets in and out of memory

Scheduling Resources

Overview

In any computer, the CPU is a limited resource. In most cases, the CPU can execute only one instruction at a time. A major function of the OpenVMS operating system is scheduling the CPU for the large number of processes that must use it.

Topics

In this section the following topics are discussed:

- Interrupts
- Interrupt priority levels
- Scheduler
- Scheduling priority
- Process states
- Dynamic priority adjustment

Interrupts

Methods of Scheduling

The CPU in any computer is a limited resource. Even though it may execute millions of instructions per second, it can only operate on one instruction at a time.

In a complex system such as OpenVMS, there may be hundreds of processes that need to use the CPU at a given moment.

There are two major methods of scheduling the use of the CPU:

- Polling
- Interrupting

The OpenVMS operating system does NOT use polling.

Interrupting

In an interrupt-driven system, a process that requires the CPU notifies the CPU immediately. If the CPU is available for use, it schedules the interrupting process.

By assigning a measure of importance to each interrupting process, the CPU can do work for processes that have a higher priority first, then move on to those processes that are less critical.

The OpenVMS operating system is interrupt driven.

Interrupt Priority Levels

Types of Priority

The OpenVMS operating system defines two types of priority:

- Software priority (scheduling priority)
- Interrupt priority level (IPL)

Interrupt Priority Level

An OpenVMS system responds to interrupts by external events. Interrupt priority level (IPL) is a measure of the relative importance of such events.

The CPU spends most of its time executing processes at IPL 0.

A process sometimes runs at an elevated IPL while executing a system-service routine. The CPU also enters an elevated IPL in response to an interrupt.



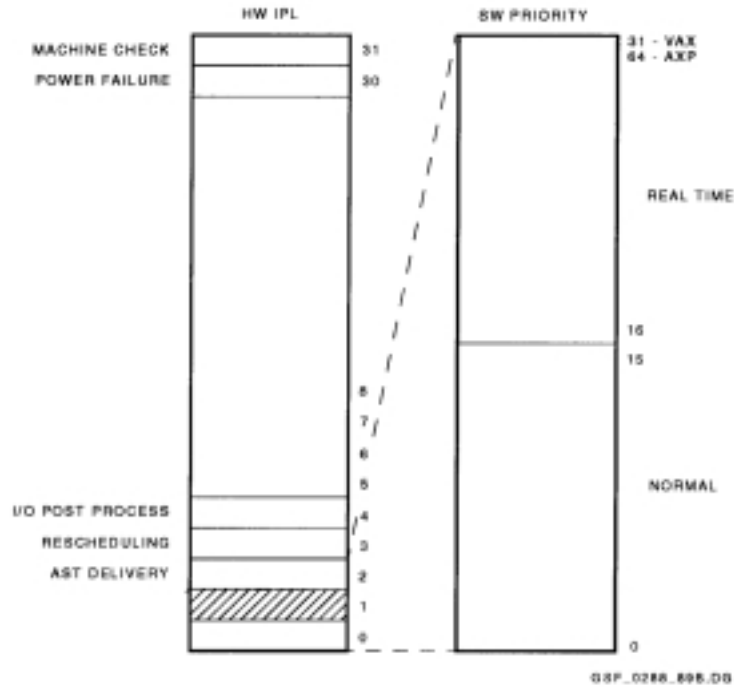
Note

During system operation, events occur that require the execution of software other than the current thread of execution. The processor responds to such events by altering the control flow from the current thread of execution. Some of these events are unrelated to the current thread and are asynchronous to it; these events are called interrupts. Other events, called exceptions, are triggered by the current thread and are synchronous to it. An example of an interrupt is a request for services for an external device.

But who gets to interrupt whom? To permit arbitration among concurrent interrupt requests and their servicing, each interrupt request has an associated interrupt priority level (IPL). When an interrupt is granted, the processor IPL is raised to that of the interrupt. A second interrupt request at that IPL level or lower is blocked from interrupting until the first is finished.

IPL and Software Priority

The interrupt priority levels (IPL) are related to the software priority. The following figure shows the relationship between IPL and software priorities:



Note

The figure shows the relationship between IPL and software (SW) priorities. It might be the student's first exposure to these ideas.

Code running in response to an interrupt runs at an elevated IPL and preempts the code running in the current process. Explain that the first responsibility of the OpenVMS operating system is to service the hardware; it runs user processes when there are no interrupts to handle.

Different IPLs are typically used for different things. The Monitor utility does not show how much time the system spends at each IPL, but the POLYCENTER Performance Data Collector for OpenVMS does.

Scheduler

The goal of the scheduler is to ensure that the CPU is always used by the highest-priority process that is ready to compute.

The scheduler:

- Selects programs to run on the CPU
- Manages process state transitions
- Manipulates the priority of time-sharing processes

Time Slice

The scheduler shares the CPU resource by dividing processor time into time slices called **quantum**. A process is selected to compute and given one time slice to execute.

A process uses the CPU until:

- It goes into a wait state.
- Its quantum expires.
- It is preempted by a software or hardware interrupt.
- It is preempted by a higher-priority process.
- It is deleted, either voluntarily or involuntarily.

Process Selection Criteria

The scheduler selects processes to use the CPU based on:

- State
- Priority

Invoking the Scheduler

System events that may invoke the scheduler include:

- Quantum end
- Request to schedule new process
- Process termination
- I/O completion
- `$ SET PROCESS/PRIORITY=n`
- Wake a process
- Timer request becomes due
- Process creation

Scheduling on a Multiprocessor System

On a multiprocessor system, the same system events invoke the scheduler as on a single-processor system.

A process may run on any processor.

- The highest-priority process is always guaranteed to execute on one of the processors.
- The next highest-priority processes are not necessarily guaranteed to execute at quantum end, unless they are real-time processes.



Note

Students migrating to VAX from other systems may ask if a multiprocessor can be segmented so that some processes run on one processor and some on another. For example, one processor is used to process banking transactions and the other is used for programmers. If the production processor goes down, the programmers are shut out and the remaining processor is given over to the production work. A bank could certainly cost-justify losing a day of programming to continue processing transactions. The answer is no. The OpenVMS Cluster system, however, provides this type of functionality.

You can go through an example in which the second-highest-priority process does not execute: say processes PROC4 (priority 4) and PROC5 (priority 5) are current. Waiting process PROC10 becomes computable at priority 10. If the event that makes PROC10 computable takes place on the CPU where PROC5 is executing, PROC10 preempts PROC5 while PROC4 continues to run on the other CPU. This situation lasts only briefly. When PROC4's quantum expires, PROC5 is scheduled again.

Guaranteeing the n highest-priority processes to run would have these disadvantages:

- Scheduler would take more CPU time.
 - Processes would migrate more frequently from one CPU to another, incurring fewer hits on the cache and translation buffer.
-

Processor Affinity Capability

Use of the Qualifiers

Using these new qualifiers allows you to keep track of processes running on defined CPUs. It is a manipulation of data that should be administered by a knowledgeable system manager.

This information is provided in the instructor guide only.

Qualifiers for SET PROCESS Command

There are two qualifiers used with the SET PROCESS command to help you keep track of running processes. These are valid only for the Alpha operating systems:

- SET PROCESS/[NO]CAPABILITIES[/SET=(n)][/CLEAR=(n)]
[/PERMANENT]
- SET PROCESS/[NO]AFFINITY[/SET=(n)][/CLEAR=(n)]
[/PERMANENT]



Note

- The /CAPABILITIES qualifier is mutually exclusive with the /AFFINITY qualifier.
- Replaces the BIND program in SYS\$EXAMPLES.
- Requires ALTPRI privileges.
- An example of when to use these qualifiers would be if you were leasing CPU time to a process and needed to keep a close accounting.

/CAPABILITIES Qualifier

The /CAPABILITIES qualifier allows bits in the user capability mask to be set or cleared individually, in groups, or all at once.

The /NOCAPABILITIES qualifier clears all the user capability bits currently set. This is based on the setting of the /PERMANENT qualifier. This qualifier alone indicates the target of the operations that are specified with the following secondary qualifiers:

- **/SET=(n,n,n)**
Sets user capabilities defined by position values *n*, where *n* has a range of 1 to 16.
- **/CLEAR=(n,n,n)**
Clears user capabilities defined by position values *n*, where *n* has a range of 1 to 16.

■ **/PERMANENT**

Forces operations to be performed on the permanent set as well as the current. Default behavior affects only the running image copy of the capabilities.

The secondary qualifiers can be used together. The set of user capability bits defined in the /SET and /CLEAR parameters should not overlap.

As with the other SET PROCESS command qualifiers, the bit operations occur on the current process if no /IDENTIFICATION qualifier or explicit process name parameter is specified.

Note



- Secondary qualifiers can all be used at once as long as the set of user capability bits defined in the /SET and /CLEAR parameters don't overlap.
 - DCL interface doesn't do any execution validation before changing scheduling state. It allows a process or a thread to go into a wait condition with no warning. This gives more flexibility to the interface commands, but increases the danger of hung processes.
 - The privileges to execute this command match those required by equivalent system service calls. ALTPRI is the base privilege required to make any modifications and the only privilege required to modify the current owner's process/thread. GROUP is required to make modifications in the same UIC group; WORLD is required to modify any unrelated process/thread.
 - The /IDENTIFICATION qualifier allows this command to affect individual kernel thread PIDs. Because each KTB can run separately, these commands treat them as discrete entities in terms of capabilities. Likewise, specifying a process name, or no process identifier at all, will only affect the initial KTB of a multi-threaded process. Specifying a process does not imply that all threads associated with the process are affected.
-

/AFFINITY Qualifier

This qualifier allows bits in the affinity mask to be set or cleared individually, in groups, or all at once. The /NOAFFINITY qualifier clears all affinity bits currently set. This is based on the setting of the /PERMANENT qualifier.



Note

- This qualifier is mutually exclusive with /CAPABILITIES.
 - Secondary qualifiers can all be used at once as long as the set of affinity bits defined in the /SET and /CLEAR parameters don't overlap.
 - DCL interface doesn't do any execution validation before changing scheduling state. It allows a process or a thread to go into a wait condition with no warning. This gives more flexibility to the interface commands, but increases the danger of hung processes.
 - The privileges to execute this command match those required by equivalent system service calls. ALTPRI is the base privilege required to make any modifications and the only privilege required to modify the current owner's process/thread. GROUP is required to make modifications in the same UIC group; WORLD is required to modify any unrelated process/thread
 - The /IDENTIFICATION qualifier allows this command to affect individual Kernel Thread PIDs.
-

Secondary Qualifiers for /AFFINITY

Specifying the /AFFINITY qualifier by itself has no direct effect other than to indicate the target of the operations specified by the following secondary parameters:

- **/SET=(n,n,n)**
Sets CPU affinities defined by position values *n*, where *n* has a range of 1 to 32 and is restricted to the set of currently active CPUs.
- **/CLEAR=(n,n,n)**
Clears CPU affinities defined by position values *n*, where *n* has a range of 1 to 32 and is restricted to the set of currently active CPUs.
- **/PERMANENT**
Forces operations to be performed on the permanent set as well as the current. This makes the changes permanent for the life of the process or thread. Default affects only the running image copy of the affinities.

As with the other SET PROCESS command qualifiers, the bit operations occur on the current process if no /IDENTIFICATION qualifier or explicit process name parameter is specified.

The /IDENTIFICATION qualifier allows this command to affect individual kernel thread PIDs. Because each KTB can run separately, these commands treat them as discrete entities in terms of capabilities. Likewise, specifying a process name, or no process identifier at all, will only affect the initial KTB of a multi-threaded process. Specifying a process does not imply that all threads associated with the process are affected.

Qualifier for START/CPU

The /DEFAULT_CAPABILITIES command qualifier, for Alpha systems only, makes the downtime for CPU shutdowns and restarts as transparent to the user interface as possible.

The CPU user capability bits will be initialized from SCH\$GL_DEFAULT_CPU_CAP at the first boot of the CPU into the system configuration.

The /DEFAULT_CAPABILITIES qualifier imitates the behavior of the initial bootstrap of the CPU. Use this qualifier to return the CPU to a known and consistent state.



Note

- Once the CPU user capability bits are initialized, they are retained across all future restarts for the life of the system.
- System capability bits are reinitialized to their defaults taken from SCH\$GL_DEFAULT_CPU_CAP.

This command format eliminates all previous capability (user and system) modifications for this CPU and reinitializes them with the values in SCH\$GL_DEFAULT_CPU_CAP:

```
START/CPU/DEFAULT_CAPABILITIES {CPU_ID}
```

Display for SHOW CPU Command

For Alpha systems, the display portion of the SHOW CPU command has changed to distinguish CPU affinity from:

- System capabilities
- User capabilities
- Process affinity

Because the code currently recognizes the longwords in which these entities reside, the changes are relatively cosmetic.

The new code presents the system and user capabilities for every CPU and every thread that has values different from the default process value.

The text indicates the appropriate user bit position when one of these bits is displayed. Before, it was described as unknown.



Note

The text indicates the bit user position.

Scheduling Priority

An OpenVMS process has two scheduling priorities:

- Base priority
- Current priority

Base Priority

The base priority is assigned when the process is created.

- An interactive login process obtains base priority from a field in the user authorization file.
- A batch process takes base priority from the batch queue that creates it.

Base priority ranges from 0 to n .

- 0 to 15 for time sharing or normal processes
- 16 to n for real-time processes:
- On a VAX system, the value of n is 31.
- On an Alpha system, the value of n is 63.

Current Priority

The current priority is used to determine the order of selection for CPU processing.

- The current priority is calculated from the base priority.
- When a process begins running, current and base priorities are equal.
- Current priority is adjusted automatically by the system for time-sharing processes.
- Current priority is not adjusted for real-time processes.

**Note**

Real-time processes:

- Have base priority above 15
- Are not removed from CPU by quantum expiration
- Deal with time-critical events
- Are run by privileged users

Example: a computer that monitors temperatures at intervals.

You cannot change the base priority of a batch process when you submit the process to the batch queue. However, you can change priority while the process is running.

Systems where interactive processes and batch processes share the CPU set base priority to favor the interactive process. For example:

- Interactive processes — 4
- Batch processes — 3

You cannot change current priority for a process, but you can change the base priority, which in turn affects the way the system calculates current priority.

Process States

On OpenVMS systems, a process is always in one of three states.

- Current — using the CPU
- Computable — ready to run, and waiting to gain access to the CPU
- Waiting — can be either voluntary or forced

Voluntary Wait States

A voluntary wait state is requested by a process, typically by means of a system service.

- Occur frequently, depending on application design
- Can be of long duration

Voluntary wait states include:

- LEF — wait for local event flag
- CEF — wait for common event flag
- HIB (hibernating) — wait to be wakened
- SUSP (suspended) — wait to be resumed

Involuntary Wait State

An involuntary wait state is not explicitly requested by a process. Involuntary wait states include:

- COM — wait for the CPU
- FPG — wait for free page
- PFW — wait for page fault read to complete
- COLPG (collided page) — wait for page fault read on shared page
- MWAIT — wait for mutex or miscellaneous resource

Involuntary wait states that occur frequently or have a long duration can indicate a system bottleneck.

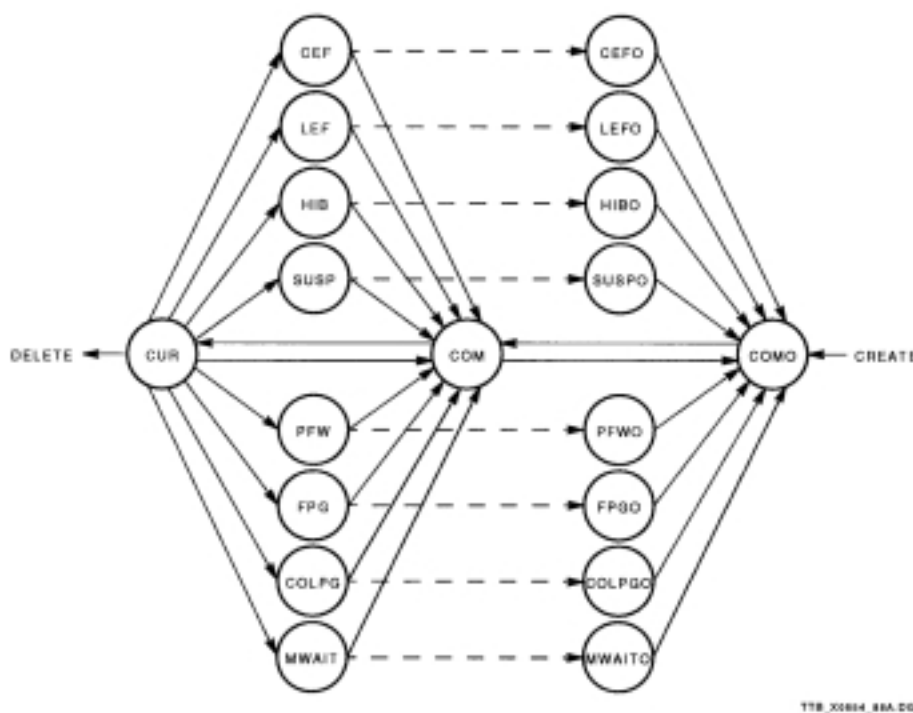
Process Behavior

Processes are characterized by their behavior.

- A compute-bound process spends most of its time in the CUR and COM states.
- An I/O-bound process spends most of its time in the LEF or CEF state waiting for I/O completion.

Figure: Process State Transition

The following figure shows process state transition:



Dynamic Priority Adjustment

Priority Boosting

The OpenVMS priority boosting scheme is designed to improve response time for processes that are not compute bound.

When a system event occurs, process priority is boosted.

- Current priority is calculated by adding the boost to the base.
- If current priority is higher than base + boost, current priority is not changed.
- Priority is never boosted beyond 15.
- A process that cannot receive the entire boost does not receive any part of the boost.
- Current priority is usually not boosted higher than the base priority plus 6.
- Boosts are not applied to real-time processes (base priority greater than 15).

System Events and Associated Priority Boosts

Different events result in different boosts in priority.

This event...	Results in this priority boost:
Process creation	6
Terminal input completion	6
Terminal output completion	4
Resource becomes available	3
Process is awakened	3
Process is resumed	3
Process is deleted	3
Process timer request ready	3
Direct I/O completes	2
Nonterminal buffered I/O completes	2
Update section write completes	2
Set process priority	2

Priority Decrementing

Each time a time-sharing process is scheduled, the scheduler decrements current priority by 1 if it is higher than the base.

Results of Priority Adjustment

The result of priority adjustment is that over time:

- Compute-bound processes operate at or near their base priority.
- I/O-bound and interactive processes tend to have average current priorities somewhat higher than their base priorities.



Note

A compute-bound process is almost always ready to process. The process uses the CPU whenever it is available and, therefore, is not hampered by the lower priority level. A compute-bound process with a high priority would freeze all other processes out.

Example: Displaying Process States and Priorities

Use the SHOW SYSTEM command to display the process states and current priorities of the processes on the system.

On VAX systems running versions of OpenVMS before Version 7.2, physical memory is displayed instead of pages.

\$ SHOW SYSTEM

OpenVMS 7.2 on node SUNDAE 15-NOV-1999 15:57:50.99 Uptime 0 18:00:41

Pid	Process Name	State	Pri	I/O	CPU	Page flts	Pages
20200081	SWAPPER	HIB	16	0 0	00:00:03.44	0 0	
20200405	MARSH	CUR	4	837 0	00:00:04.63	4195 396	
20200086	CONFIGURE	HIB	10	46 0	00:00:00.08	127 253	
20200088	IPCACP	HIB	10	46 0	00:00:00.09	115 231	
20200089	ERRFMT	HIB	8	955 0	00:00:04.05	115 230	
2020008A	CACHE_SERVER	HIB	16	117 0	00:00:00.16	77 181	
2020008B	CLUSTER_SERVER	HIB	10	10 0	00:00:01.43	137 423	
2020008C	OPCOM	HIB	7	8922 0	00:00:32.60	755 322	
2020008D	AUDIT_SERVER	HIB	10	160 0	00:00:00.93	484 890	
2020008E	JOB_CONTROL	HIB	10	820 0	00:00:02.05	226 413	
2020008F	QUEUE_MANAGER	HIB	8	3104 0	00:00:49.34	1336 1726	
20200090	SHADOW_SERVER	HIB	4	6 0	00:00:00.03	84 194	
20200091	SMISERVER	HIB	9	32 0	00:00:00.27	289 691	
20200092	SYMBIONT_1	LEF	4	318 0	00:00:02.09	614 945	
20200093	SYMBIONT_2	LEF	4	851 0	00:00:05.52	657 1689	
20200094	SYMBIONT_3	HIB	4	50 0	00:00:01.10	548 816	
20200095	SYMBIONT_4	HIB	4	10 0	00:00:00.17	279 253	
20200096	SYMBIONT_5	HIB	4	10 0	00:00:00.18	279 253	
20200097	NETACP	HIB	10	6581 0	01:01:36.38	8456 8439	
20200098	EVL	HIB	6	6965 0	00:00:08.17	351 666 N	
20200099	REMACP	HIB	9	64 0	00:00:00.07	97 126	
2020009A	LATACP	HIB	14	9 0	00:03:32.86	643 705	

.
. .
.

Note

Ask the students how many processors the system has. On a multiprocessor system, the CPU ID for each current process appears next to the State field.

Discuss the priority display and the process states.

Network Configurations

Overview

A network allows two or more computers to share resources. These resources could be such things as:

- Mass storage
- Memory
- CPU cycles

There are several general network configurations possible using Compaq hardware and software.



Note

You might want to set the stage for this section by indicating that DECnet is only one of various protocols that can run over the Ethernet. Others include TCP/IP, LAT, LAST, OpenVMS Cluster, and PATHWORKS. (The Ethernet LAN figure in this section shows several of these.)

Topics

In this section the following topics are discussed:

- Network configurations
- Network hardware
- DECnet routing
- DECnet node names

Network Configurations

DECnet Network

DECnet is the collective name for the family of communications products (software and hardware) that allow Compaq operating systems to participate in a network.

A DECnet network consists of two or more computing systems linked for the purpose of exchanging information and sharing resources.

DECnet Network Characteristics

Characteristics of a DECnet network include:

- OpenVMS systems use DECnet VAX software to participate in DECnet networks.
- All systems connected to a DECnet network are peers.
- Any system in the network can communicate with any other system in the network.
- Each system in the network is called a **node**.
- Nodes are connected by lines, over which circuits operate.
- DECnet networks can vary in size from 2 nodes to over 64,000 nodes.
- Large networks can be divided into multiple **areas**:
 - There can be up to 63 areas.
 - Each area can contain up to 1023 nodes.
 - Each area operates as a subnetwork.

DECnet Network Configurations

DECnet networks support a variety of different network configurations, including:

- Local area networks (LANs) using either:
 - Fiber Distributed Data Interface (FDDI) for high-speed communications (100 Mbit/s)
 - IEEE 802.3/Ethernet for low-cost, flexible configurations (10 Mbit/s)
- Wide area networks (WANs) utilizing point-to-point, synchronous and asynchronous connections, as well as multipoint connections
- Communication through packet switching networks
- Communication with IBM SNA networks



Note

Systems can communicate with each other without having to go through a central or master system.

It is recommended that any network with more than 500 nodes be subdivided into areas. The reasons for dividing a network may include:

- Technical
 - Geographical
 - Managerial/political
-

DECnet Node Names

Naming conventions for DECnet node names correspond to the two types of DECnet functionality:

- DECnet for OpenVMS node names

These names are used in the default version of DECnet shipped with the OpenVMS operating system. Refer to the *DECnet for OpenVMS Networking Manual* for more information.

- DECnet/OSI full names

Full names are hierarchically structured DECnet node names that can be stored in a DECdns naming service. Full names can be a maximum of 255 bytes long.

Full Names Syntax

Full names have the following general syntax:

namespace:.directory directory.node-name

where:

- *namespace* identifies the global naming service
- *directory directory* defines the hierarchical directory path within the naming service
- *node-name* is the specific object defining the DECnet node

The node full name must begin with the namespace, followed by a colon (:). The directory path must begin with a dot (.).

Example: Full Names Syntax

In the following example of a full name, OMNI is the name of a namespace in a global naming service; .US.Massachusetts.Boston is the directory path to the object, RUBY, which represents the node.

OMNI:.US.Massachusetts.Boston.RUBY



Note

The system stores a full name as you enter it, preserving uppercase and lowercase entries. However, when matching an entry with a stored full name, the system is case insensitive; in other words, if the user enters Omni, the system recognizes it as equivalent to OMNI.

Rules for Assigning Full Names

Consider the following when assigning node names:

- Node full names must be unique within your network.
- OpenVMS VAX systems support any type of node name except one containing an odd number of double quotes.
- You must place quotes (" ") around a node name under any of the following conditions:
 - The node name contains a space, tab, comma, left parenthesis, right parenthesis, single quote ('), double quote ("), slash (/), exclamation point (!), plus sign (+), or at sign (@).
 - The node name contains the character sequence of two colons (::).
 - The node name starts or ends with a colon (:) character.
- If a DECnet node name contains double quotes, you must duplicate each double quote. Also, you must be sure to pair double quotes within a full name. For example, the node name *foo:"bar"* must appear as *foo:""bar""*.

Although the OpenVMS operating system enforces few rules on the syntax of node names, the DECnet for OpenVMS software running on your system limits the actual set of valid node names you can use.

For more information on full names, refer to your DECnet/OSI documentation.

Network Hardware

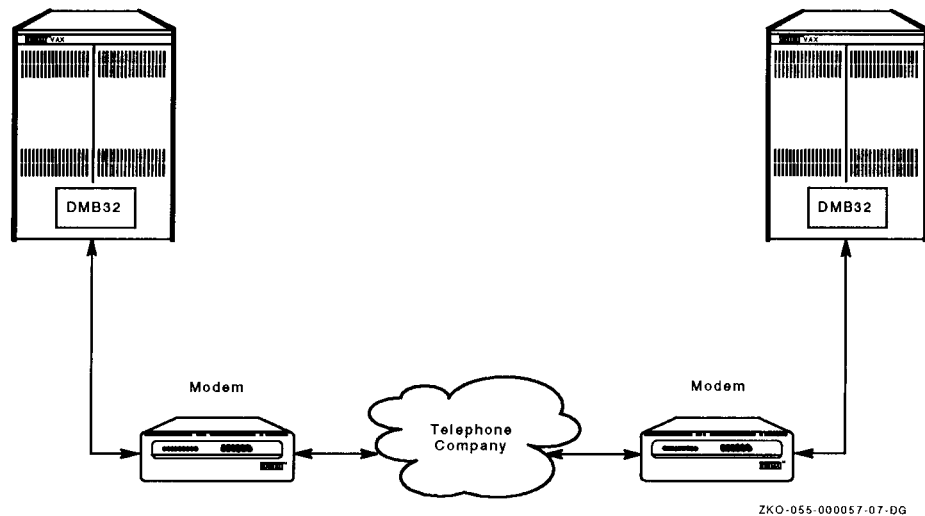
A DECnet network can comprise many different types of hardware.

Point-to-Point Connections

- A point-to-point connection uses WAN technology to link two nodes. WAN technology utilizes:
 - A synchronous or asynchronous communication interface.
 - Synchronous communication is preferred when large amounts of data must be transmitted on a continual basis.
- Asynchronous communication is advantageous when a temporary or low-cost connection is needed.
- A modem to modulate the signal from digital to analog for transmission over the communication line, and to demodulate the signal from analog to digital at the receiving end.
- A communication cable, usually a telephone line.

Figure: Point-to-Point Connection

The following figure shows a point-to-point connection:



FDDI LAN Standard

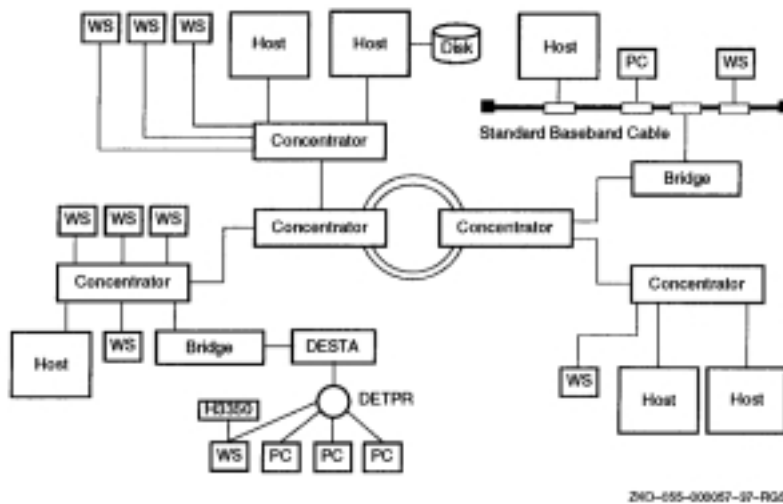
- FDDI (Fiber Distributed Data Interface) is the first LAN standard optimized for fiber optics.
- Operates at 100 Mbit/s
- Can operate as a dedicated channel or serve as a high-speed backbone connecting multiple 802.3/Ethernet subnetworks
- Can operate simultaneously as an OpenVMS Cluster interconnect
- Supports a maximum of:
 - 500 network stations (concentrators or bridges)
 - Ring circumference of 100 km (62 miles)
 - Distance between stations of 2 km (1.2 miles) using multimode fiber-optic cable or 40 km (25 miles) using single-mode fiber-optic cable

Note

FDDI can also run on Thin Wire or shielded twisted-pair (STP) cable, allowing the use of existing wiring to connect desktop systems to the fiber backbone.

Figure: FDDI Network

The FDDI ring is in the center of the following figure:



802.3/Ethernet LAN Standard

Compaq supports the IEEE 802.3/Ethernet standard, the most widely implemented standard for local area networking.

- Operates at 10 Mbit/s.
- The longest distance between any two stations on an Ethernet LAN is 2800 meters (9187 feet).
- If a segment is longer than 500 meters (1641 feet), there must be a **repeater** to refresh and regenerate the signal.
 - Repeaters contain no intelligence. They retransmit every bit of information they receive, regardless of the location of the source and destination nodes. They provide no filtering.
 - Local repeaters can span up to 100 meters (328 feet).
 - Remote repeaters connect two repeaters with fiber-optic cable to span up to 1000 meters (3281 feet).
 - No more than two repeaters are permitted between any two stations. A remote repeater is considered one repeater.
- An extended Ethernet LAN supports up to 8,000 stations.
- An 802.3/Ethernet configuration can support several other networking protocols simultaneously with DECnet protocols. In particular, it can serve as an OpenVMS Cluster interconnect.
- Ethernet can run over baseband coaxial cable, broadband coaxial cable, ThinWire, and twisted pair.
- The Ethernet protocol is the same regardless of the physical medium it runs over; however, the configuration rules are different for standard Ethernet and ThinWire.

The following diagram shows how the 2800 meter (9187 feet) protocol limit is reached (moving from A to B):

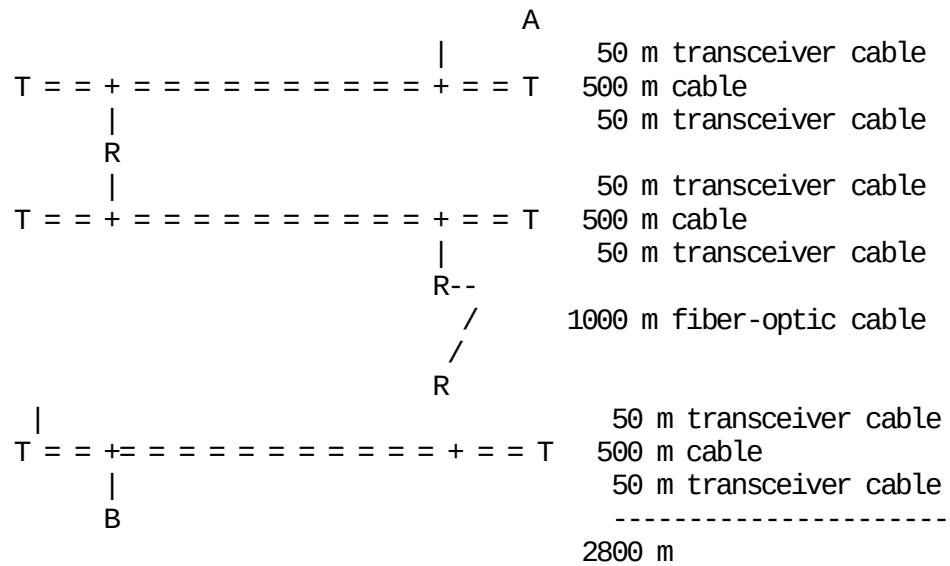


Figure: Ethernet LAN

This figure shows two non Compaq workstations attached to the same Ethernet LAN that is running DECnet software. These two can use the same LAN to communicate using TCP/IP. There are also several terminal servers attached to the Ethernet; these operate using the LAT protocol.

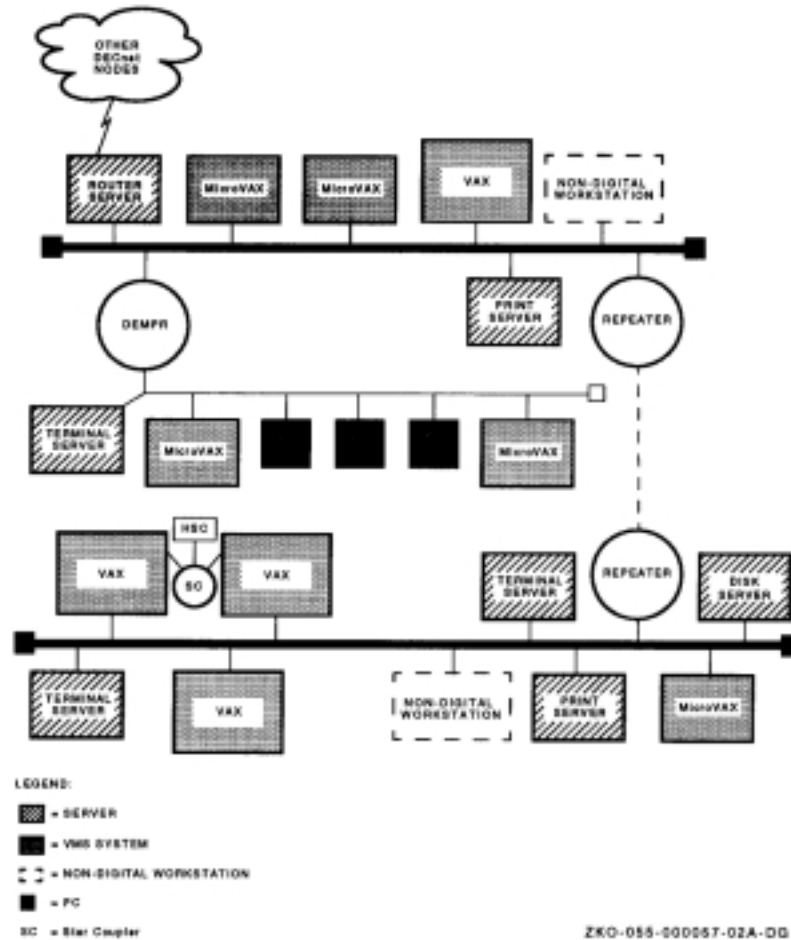


Figure Notes: Ethernet LAN

- Several protocols can run over Ethernet simultaneously. This figure shows at least four different protocols, all communicating over the same Ethernet LAN.
 - DECnet
 - TCP/IP running between non Compaq workstations running the UNIX operating system
 - LAT running between terminal servers and hosts
 - LAST (local area services transport) running between disk server and hosts, and between hosts running PATHWORKS software and their PC clients
- Ethernet can run on baseband, broadband, or ThinWire cable. This figure shows baseband and ThinWire.
- PCs running DOS can run DECnet-DOS to participate in a DECnet network.
- Note that both nodes in the CI (star coupler) cluster also have Ethernet connections. It is not a good idea to rely on the CI to run DECnet. The CI should be used as a backup circuit only.
- The disk server is a network node.



Note

You might want to mention the DELNI here even though it is not shown in the figure. The DELNI is an “Ethernet in a box”.

A single DELNI can connect up to 8 stations.

You can cascade DELNIs so that a DELNI is attached to each of the 8 ports. In this way, you can connect up to 64 stations.

A noncascaded DELNI can connect to a standard Ethernet. This is a cost-effective and space-saving option when several machines are very close to each other.

Servers

A **server** is an entity that is responsible for processing and fulfilling requests. The issuer of these requests is called the requester.

On a LAN, a server is usually a special-purpose node that provides a service to the other nodes on the LAN.

This kind of server...	Performs these functions:
Terminal server	• Provides a service to both the user and the host node. • Eliminates the need for terminals to be hardwired to hosts. • Allows any terminal connected to a terminal server to establish a connection with any host on the LAN. • Allows users to establish multiple simultaneous connections to the same or different hosts.
Print server	• Provides a printing service to the network. • Allows a high-speed, high-quality printer to be shared by several nodes, thus distributing the expense.
Disk server	• Provides disk services to the network. • Allows a disk drive to be shared by several nodes, thus distributing the expense.
Router server	• Provides a routing service to nodes on the LAN. • Permits nodes on the LAN to communicate with other DECnet nodes that are not on the LAN. • Can improve the efficiency of routing on the LAN.



Note

A server is any network node where shared resources are located. A computer in a LAN that provides services to the other nodes can be viewed as providing server functions (whether or not the node is dedicated to those functions). A host providing server functions can run other applications if its nonprocessing services do not justify total dedication.

Adding servers to the network to handle nonapplications-processing functions frees up processing time on other hosts to handle user applications.

Extended LANs

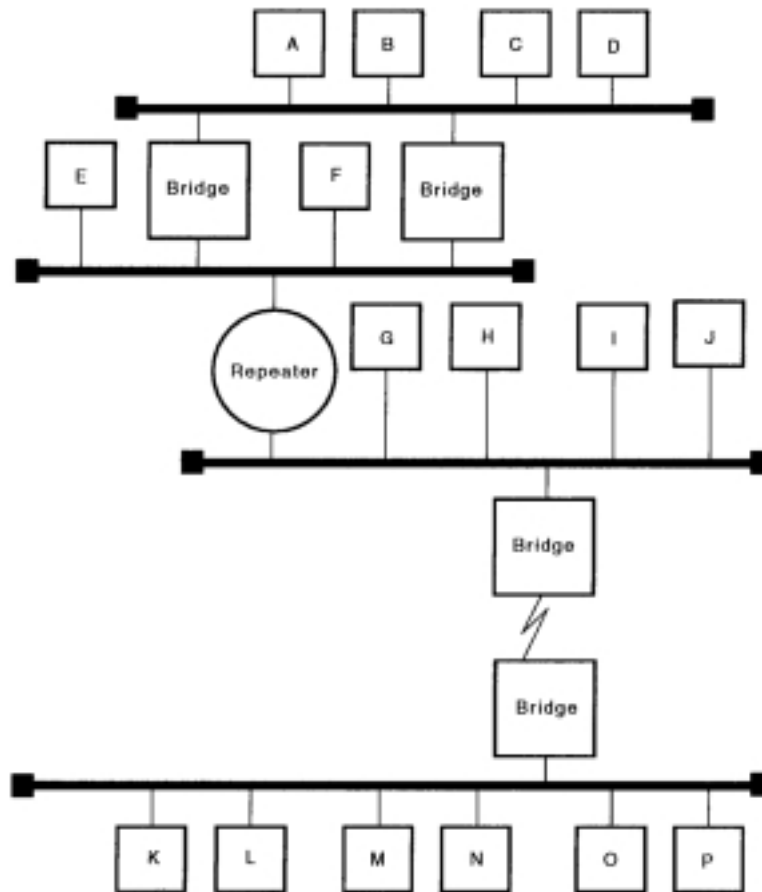
Extended LANs are created by using several different types of **bridges**. A bridge is a device used for connecting two segments of a LAN.

Bridges can be used to:

- Join two or more LANs to create an extended LAN.
- Connect one type of LAN to another (802.3/Ethernet to FDDI).
- Join LANs in remote locations to provide LAN capabilities where an ordinary LAN cannot reach.
 - In an 802.3/Ethernet LAN, a remote repeater can extend the network by only 1000 meters (3281 feet).
 - With fiber-optic and microwave bridges, an extended LAN can span distances of up to 22,000 meters (72,182 feet) and include up to 8,000 nodes.
 - There are bridges available that use satellite technology to span even greater distances. The throughput for a satellite bridge is limited due to propagation delays.
- Improve performance on heavily used LANs.
 - Repeaters forward all data packets, including those where the destination node is on the same segment as the source node.
 - Bridges filter data and forward only those packets where the destination node is on a different segment than the source node.

Figure: Extended LAN

The following figure shows an extended LAN configuration:



ZKO-055-050057-14-DG



Note

Point out the difference between a bridge and a repeater, as shown in the figure.

When there are two or more bridges between two Ethernet LANs, only one performs the store and-forward functions. The other can be used as a backup or as a monitor. (In fact, the LAN Traffic Monitor requires a bridge to be used solely as a monitor.)

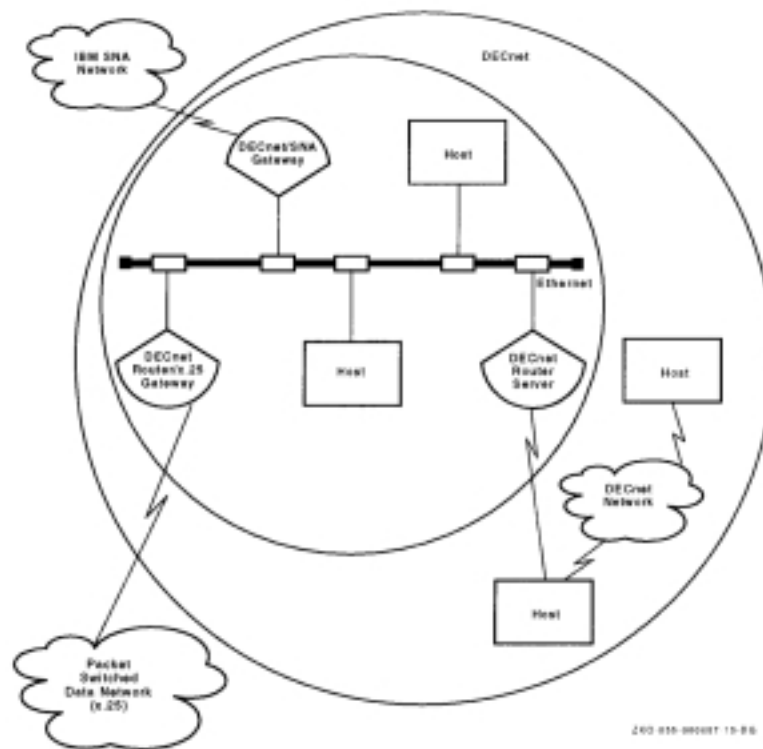
Multivendor Networks

DECnet software is flexible enough to participate in multivendor networks. To do so requires a **gateway**, as well as special software on the OpenVMS system, to communicate with the other networks. A gateway is a module, or set of modules, that transforms the conventions of one network into the conventions of another.

- The function of a gateway is to translate protocols between dissimilar systems.
- The DECnet SNA gateway allows DECnet systems on the LAN to communicate with IBM SNA networks.
- The DEC X25router and DEC X25gateway products allow DECnet systems on the LAN to participate in packet-switched data networks that use the X.25 protocol.

Figure: DECnet Software in a Multivendor Environment

The following figure shows DECnet software in a multivendor environment:



DECnet Routing

Routing

If two nodes that are not directly connected need to communicate, the DECnet software routes information between them.

How Routing Works

To route information between nodes, the DECnet software:

- Selects the best (least-cost) path between the source and destination nodes.
- Provides adaptive routing. If the preferred path is unavailable, the software selects an alternate path.

If two paths have the same cost:

- When equal-cost path-splitting is enabled, packets alternate between equal paths.
- When the maximum number of path splits is one, packets are sent along the shorter path.

The DECnet software performs this function automatically, provided the network manager sets up **routing nodes** at the correct locations.



Note

Students need to understand what routing is because when they learn to run NETCONFIG.COM, they have to know whether to configure a routing node or an end node.

Circuit versus line: the line is defined by the hardware and the circuit is defined by the software.

Routing Terminology

The following terms are used in routing:

- **Line** — A physical channel between nodes.
- **Adjacent nodes** — Nodes that have a line connecting them.
- **Circuit** — The logical channel between adjacent nodes; multiple circuits can share a line (such as over CI).
- **Hop** — The logical distance to an adjacent node.
- **Circuit cost** — A positive integer associated with using a circuit.
- **Path** — A possible route from source node to destination node.
- **Path length** — The total number of hops between source and destination nodes.
- **Path cost** — The sum of circuit costs between source and destination nodes.
- **Path splitting** — The ability of DECnet software to divide the load over multiple paths, if multiple paths exist between a pair of nodes.

Node Types

There are two types of nodes in a DECnet network:

- **End node**
 - Has exactly one active circuit
 - Can communicate only through its adjacent node
- **Full-function (routing) node**
 - Can have more than one active circuit
 - Receives and forwards information addressed to other network nodes
 - Exchanges information about network availability with other routing nodes

Area Routing

A DECnet network can comprise over 64,000 nodes. To make the network more manageable, it can be divided into areas.

- An area can contain up to 1023 nodes.
- A DECnet network can contain up to 63 areas.
- In a network with multiple areas, there are two types of routing nodes:
 - Level 1 routers that can route data within an area
 - Level 2 routers that route data between areas
- When there are two or more area routers in an area, they should form a closed loop.

**Note**

There are really three node types if you count level 1 and level 2 routers as separate types. There are only two types of DECnet software license.

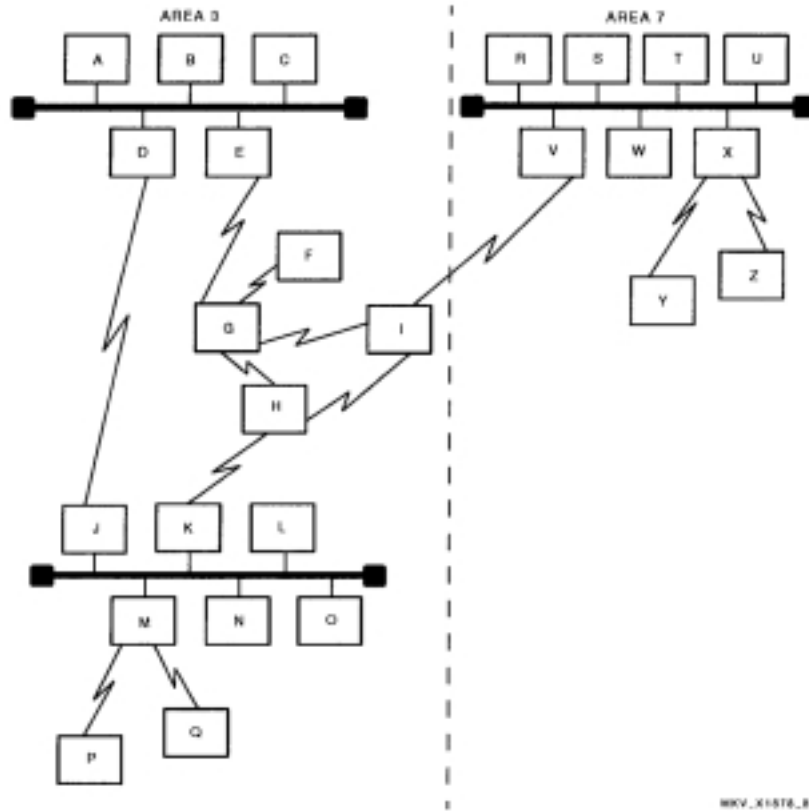
Different node types maintain different kinds of routing databases:

- End node – no routing database
 - Needs no database because it does no routing
 - Knows only about its adjacent node
- Level 1 routing node – one routing database
 - Stores information about all nodes within its area
 - Keeps track of the nearest area router (level 2) to route data outside of the area
 - Exchanges routing information about nodes within its area with all adjacent routing nodes
- Level 2 routing node – two routing databases
 - Stores information about all nodes within its area
 - Exchanges routing information about nodes within its area with all adjacent level 1 routing nodes
 - Stores information about all other level 2 routers

Exchanges routing information about other level 2 routers with all adjacent level 2 routing nodes

Figure: DECnet Node Types

The following figure shows end nodes and routing nodes in a DECnet network. Any node with two or more lines attached to it is a routing node.



Routing on an Ethernet LAN

Routing on an Ethernet LAN works in the following manner:

- All nodes on an Ethernet LAN are adjacent.
- Routing nodes on an Ethernet LAN:
 - Allow Ethernet nodes to communicate with nodes that are not on the LAN
 - Keep track of the availability of other nodes
- The designated router intervenes in transmissions between end nodes.
- Router servers and other dedicated routing nodes can further enhance the performance of an Ethernet LAN by:
 - Reducing or eliminating routing traffic on the LAN
 - Allowing individual nodes to spend less CPU time on routing

**Note**

There can be too many routers on an Ethernet LAN. Routing nodes send periodic update messages. These messages create a lot of traffic on the Ethernet. Excess traffic on an Ethernet LAN causes more collisions and less efficient transmission. Therefore, the majority of nodes on the Ethernet should be end nodes.

If there are no routers on an Ethernet LAN:

- End nodes exchange messages directly.
- Data is broadcast to the entire Ethernet, and each node recognizes its own address.

If there is one router on the Ethernet, it intervenes in transmissions between end nodes and:

- Provides a routing service for end nodes on the Ethernet.
- Eliminates traffic destined for unreachable nodes by notifying end nodes when another end node on the Ethernet is not available.

When there is more than one router on the Ethernet, the router with the highest router priority is selected as the designated router. If there is a tie for the highest router priority, the node with the higher address is selected.

It is important to select a specific router to be the designated router because the choice of designated router affects network performance.

Summary

Concepts

Virtual Memory Management Concepts

- A virtual memory management system allows the execution of images that may not be entirely in physical memory.
- Virtual addresses are mapped to physical pages by means of a page table.
- The virtual address space is divided into system space and process space.

Working Sets

- A working set is the set of physical pages currently being used by a process.
- Important working set parameters include:
 - WSDEFAULT
 - WSQUOTA
 - WSEXTENT

Paging and Swapping

- Paging occurs when a program needs access to a location not currently in physical memory.
- Swapping occurs when an entire image is moved to disk to free up physical memory.
- The SWAPPER process is an integral part of the OpenVMS operating system.

Scheduling Resources

- The OpenVMS operating system is interrupt driven.
- Interrupt priority level (IPL) is a measure of the relative importance of events that interrupt the system.
- The scheduler ensures that the CPU is always used by the highest-priority process that is ready to compute.
- The priority of a process is used to determine the order of selection for CPU processing.
- A process can be in one of three states:
 - Current
 - Computable
 - Waiting
- The OpenVMS operating system uses dynamic priority adjustment to improve response time for processes that are not compute bound.

Network Configurations

- DECnet networks support a variety of configurations.
- Network configurations can include a great variety of hardware components.
- If two nodes that are not directly connected need to communicate, DECnet software routes information between them.

Managing Batch and Print Requests

Module 2

About This Module

Introduction

The OpenVMS operating system provides comprehensive facilities to dynamically manage print and batch requests submitted to the system.

This module discusses how to create, manage, and control queues and devices used to handle batch and print requests.

Objectives

To manage batch and print requests on an OpenVMS system, you should be able to:

- Manage queues and spooled devices on an OpenVMS system.
- Control batch and print jobs.

Managing Queues and Spooled Devices

Overview

When a user submits a batch or print job by using the DCL SUBMIT or PRINT commands, the job is sent to a queue for processing. When the user selects the print option in some applications, the output is sent directly to an output device for processing. The OpenVMS operating system has methods to help manage both kinds of requests.

When submitting jobs to be scheduled through queues, information about the user's queue request; such as type of job, file names, queue names, and special options, is sent to the queue manager. The queue manager then retrieves information from the queue database to identify the most suitable queue for the job. The job is then placed in the appropriate queue to await processing.

When submitting data through applications, the application will allocate the selected device for the entire time the application is running, unless the system manager has spooled the device to allow other users access.

The system manager is responsible for creating and defining queues to handle the different types of job requests that users on a system will make.

Topics

In this section the following topics are discussed:

- OpenVMS Management Station
- Creating execution queues
- Creating generic queues
- Deleting queues
- Naming queues
- Managing queue characteristics
- Creating batch queues
- Creating print queues
- Creating additional queue managers
- Creating queues under additional queue managers
- Setting up logical ports
- Setting up spooled devices

OpenVMS Management Station

OpenVMS Management Station is a Microsoft Windows and Windows NT based management tool for system managers and others who perform management tasks on OpenVMS systems. OpenVMS Management Station provides a comprehensive user interface to OpenVMS account, printer, and storage management.

OpenVMS Management Station allows you to organize the systems you need to manage in ways that are meaningful to you and your environment, and allows you to manage user accounts, printers, and storage on those systems.

OpenVMS Management Station consists of two components. You install the client software on a PC to perform all management operations. You install the server component on all of the OpenVMS systems you want to manage. You do not interact directly with the server component.

Printer Management

OpenVMS Management Station makes it easy for you to manage a wide range of printers and print queues across multiple OpenVMS Cluster systems and OpenVMS nodes. In addition, the printer monitoring feature allows you to quickly detect and correct printer problems.

You do not need to maintain complicated command files to control your printer environment. You can create, delete, and manage a printer and its related queues, as well as perform print job management for those printers, from a Windows interface.

Some of the tasks you can perform include:

- Monitor one or more printers
- Examine and modify printer attributes
- Delete printers and their associated queues
- Create printers and their associated queues
- Examine and modify queue attributes
- Examine and modify job attributes, requeue jobs, delete jobs

Printer Management Operations

OpenVMS Management Station supports the following printer management operations:

For this task	Choose
Create a printer and its queues	Create from the Action menu or right-click options
Monitor printers	Monitor from the Action menu or right-click options
Modify printer, queue, and job attributes	Properties from the Action menu or right-click options
Delete a printer and its queues	Delete from the Action menu or right-click options
Rename a printer	Rename from the Action menu or right-click options
Stop, start, or restart a printer or queue	Stop, Start, or Restart from the Action menu or right-click options

Creating Execution Queues

Execution Queues

Execution queues accept batch and print jobs for processing.

- Batch execution queues contain one or more detached processes that run command procedures.
- Output execution queues accept print jobs for processing by an independent process called an output **symbiont**. This output symbiont process transfers data from a disk to an output device, such as a printer.

Creating an Output Execution Queue

To create an output (device) execution queue, use the DCL INITIALIZE command.

Format:

INITIALIZE/QUEUE/DEVICE [/qualifiers] queue-name

Example: Creating an Output Execution Queue

The following is an example of using the INITIALIZE command to create an output execution queue:

```
$ INITIALIZE/QUEUE❶/START❷ /DEVICE=PRINTER LPA0:
$ INITIALIZE/QUEUE/START❸ /DEVICE=TERMINAL TTA3:
$ INITIALIZE/QUEUE/START/DEVICE=PRINTER❹ /ON=LPA0: ❺SYSS$PRINT
```

Example Notes: Creating an Output Execution Queue

Notes on the Creating an Output Execution Queue example include:

- ❶ /START starts the queue.
- ❷ /DEVICE=PRINTER creates a symbiont process for this printer queue.
- ❸ /DEVICE=TERMINAL creates a symbiont process for this terminal queue.
- ❹ /ON=LPA0: associates the created queues with printer LPA0:.
- ❺ SYSS\$PRINT is the default system printer queue.

Creating Generic Queues

Generic Queues

Generic queues hold jobs until they can be transferred to an execution queue for processing.

- Generic batch queues can be used to distribute the workload across several nodes in an OpenVMS Cluster system.
- Generic execution queues direct output to one of three types of output device: printer, terminal, or server. When you define a generic execution queue, you must associate it with one of these types of device.

Before you can create a generic queue, you need to create an execution queue for processing the job.

Creating a Generic Batch Queue

A batch queue can execute more than one job at a time.

- On single node systems, you may have to create more than one batch queue and assign them all to a batch execution queue.
- On an OpenVMS Cluster system, you can use batch execution queues to distribute batch processing across several nodes in the cluster.

To create a generic batch queue, include the /GENERIC qualifier to the INITIALIZE/QUEUE command.

Format:

INITIALIZE/QUEUE/BATCH/GENERIC=(queue1,...) queue

- To set up a clusterwide batch execution queue, create a clusterwide queue name it SYS\$BATCH.
- The default is not to create a generic queue.

Sample Commands for Creating Generic Batch Queues

The following command creates and starts a generic queue, SYSS\$BATCH, which will feed any batch execution queues that have generic feeding enabled.

```
$ INITIALIZE/QUEUE/BATCH/START/GENERIC SYSS$BATCH
```

The following command creates and starts a generic queue, SYSS\$BATCH, which will feed the execution queues WIND\$BATCH and ICE\$BATCH.

```
$ INITIALIZE/QUEUE/BATCH/START -  
_ $ /GENERIC=(WIND$BATCH, ICE$BATCH) SYSS$BATCH
```

Creating Generic Print Queues

Generic print queues are created in the same way that generic batch queues are created. Use the /GENERIC qualifier to the INITIALIZE/QUEUE command.

The following command creates and starts a generic queue, SYSS\$PRINT, which will feed any print execution queues that have generic printing enabled.

```
$ INITIALIZE/QUEUE/START/GENERIC SYSS$PRINT
```

The following command creates and starts a generic queue, SYSS\$PRINT, which will feed the execution queues LPA0 and LPB0.

```
$ INITIALIZE/QUEUE/START/GENERIC=(LPA0, LPB0) SYSS$PRINT
```

Deleting Queues

Deleting Queues

Deleting a queue removes it from the running system, and removes the queue definition from the queue database. You should make certain that any job running on the queue is stopped before deleting the queue.

Use the DELETE/QUEUE command to delete a queue.

Format:

DELETE/QUEUE queue

Example: Deleting a Queue

The following example shows how to delete a queue:

```
$ SHOW QUEUE BIG
Batch queue BIG, busy, on BAILEY::
  Entry  Jobname  Username  Status
  ----  -
❶  321  LOOPS    DUNHAM    Executing
❷  $ STOP/QUEUE/NEXT BIG
    $ SHOW QUEUE BIG/ALL
❸  Batch queue BIG, busy, stop pending, on BAILEY::
  Entry  Jobname  Username  Status
  ----  -
    321  LOOPS    DUNHAM    Executing
❹  $ STOP/QUEUE/RESET BIG
    $ SHOW QUEUE BIG/ALL
❺  Batch queue BIG, stopped, on BAILEY::
    $ DELETE/QUEUE BIG
    $ SHOW QUEUE BIG
    %JBC-E-NOSUCHQUE, no such queue
    $
```

Example Notes: Deleting a Queue

Notes on the Deleting a Queue example include:

- ❶ The queue BIG has a job executing.
- ❷ STOP/QUEUE/NEXT waits for the current job to end, then stops the queue; this is a polite way to delete a queue.
- ❸ The queue is in the **stop pending** state.
- ❹ In this case, the job is in an endless loop; the /RESET qualifier abruptly terminates the job.
- ❺ The queue is in the **stopped** state and can be safely deleted.

Naming Queues

Queue Names

The default name for any print execution queue is the name of the physical device associated with it. To make use of generic and execution queues, you may have to name queues in such a way that the name indicates the usage or limitations of the queue.

Naming a Queue Using the Default Value

To name a queue using the default physical device name, use the following format:

INITIALIZE/QUEUE/START device-name:

In the following command, LPA0: is the physical device name.

```
$ INITIALIZE/QUEUE/START LPA0:
```

Naming a Print Execution Queue

To create an execution queue and assign a name to it that describes the queue's function, use the following format:

INITIALIZE/QUEUE/START/ON=device-name: queue-name

In the following example, LPC0: is a physical device being associated with the queue named MEMO.

```
$ INITIALIZE/QUEUE/START/ON=LPC0: MEMO
```


Managing Queue Characteristics

Queue Characteristics

The system manager can define characteristics and assign them to queues.

Users can include the /CHARACTERISTIC qualifier with a PRINT or SUBMIT command. All the characteristics the user specifies must also be specified for the queue. If there is a mismatch, the job remains pending (not executing) until the characteristics match. Users do not need to specify all characteristics of a queue, just a subset.

To manage queue characteristics, the system manager can:

- Define characteristics
- Assign characteristics to queues
- Display characteristics assigned on a system
- Display characteristics assigned to a queue
- Delete characteristics



Note

Suppose a system manager manages 12 LN03 printers, three of which are located in each of the four corners of a building. A generic queue LN03\$PRINT feeds execution queues for each of the LN03 printers.

The system manager defines four characteristics, EAST, WEST, NORTH, and SOUTH. The EAST characteristic is assigned to the execution queues that feed printers in the eastern corner of a building. WEST is assigned to the execution queues feeding the printers in the western corner, and so on. When a user submits a print job to LN03\$PRINT and specifies the EAST characteristic, the job prints on the first idle LN03 printer in the eastern corner of the building.

If the system had queues for printers on multiple floors, the system manager could further define a characteristic for each floor, for example; FIRST, SECOND, and THIRD.

Defining Characteristics

By default, no queue characteristics are defined. To define a characteristic, use the `DEFINE` command.

Format:

DEFINE/CHARACTERISTIC *char-name* *char-number*

- The *char-name* is the name of the characteristic.
- The *char-number* is the number associated with that characteristic.

Example: Defining Queue Characteristics

The following command defines the characteristic `EAST` with an associated number of 1:

```
$ DEFINE/CHARACTERISTIC EAST 1
```

Once this definition is created, the characteristic `EAST` can be associated with, for instance, all printers on the east side of a building. Users desiring to have output directed to printers in that area would specify the characteristic `EAST` in their print command.

Example: Displaying Characteristics Defined on a System

To display the characteristics defined on a system, use the `SHOW QUEUE/CHARACTERISTIC` command.

```
$ SHOW QUEUE/CHARACTERISTIC
```

Characteristic name	Number
-----	-----
EAST	1
WEST	2
NORTH	3
SOUTH	4

Assigning Characteristics to a Queue

By default, no characteristics are assigned to a queue. To assign one or more characteristics to a queue, include the /CHARACTERISTICS qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command.

Format:

SET QUEUE/CHARACTERISTICS=char[...] queue-name

Example Assigning Characteristics to a Queue

The following example uses SET QUEUE/CHARACTERISTICS to assign characteristics to a queue:

```
$ SET QUEUE/CHARACTERISTICS=EAST LN03_1
```

Users specifying EAST in a print command would have output directed to LN03_1 or any other printer with the EAST characteristic.

Example: Displaying Characteristics Assigned to a Queue

To determine the characteristics defined for a queue, use the SHOW QUEUE command.

```
$ SHOW QUEUE/FULL LN03_1
```

Printer queue LN03_1, idle, on HERA::TTA3, mounted form DEFAULT

```
<Printer queue on node HERA for an LN03 printer>
/BASE_PRIORITY=4 ❶/CHAR=(1)
/DEFAULT=(FLAG=ONE,FORM=LN03$PORTRAIT
(stock=DEFAULT)) /LIBRARY=LN03LIBRARY Lowercase
/OWNER=[SYSTEM] /PROCESSOR=LATSYM
/PROTECTION=(S:E,O:D,G:R,W:W)
/SEPARATE=(RESET=(ANSI$RESET))
```

Example Notes: Displaying Characteristics Assigned to a Queue

Notes on the Displaying Characteristics Assigned to a Queue example include:

- ❶ The characteristic displayed is number 1.

When the EAST characteristic was defined, the system manager assigned it a number of 1. You can determine what numbers represent what characteristics by using the SHOW QUEUE/CHARACTERISTIC command.

Canceling Characteristics Assigned to a Queue

To cancel characteristics assigned to a queue, you can either remove all characteristics from the queue using the SET QUEUE /NOCHARACTERISTICS command, or redefine the queue characteristics using the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command.

Format:

```
SET QUEUE/NOCHARACTERISTICS queue-name  
START/QUEUE/NOCHARACTERISTICS queue-name  
INITIALIZE/QUEUE/NOCHARACTERISTICS queue-name  
SET QUEUE/CHARACTERISTIC=char queue-name
```

Whenever you reset queue characteristics, you automatically cancel any characteristics set by a previous INITIALIZE or SET command.

Deleting Characteristics

To delete a characteristic definition, use the DELETE command.

```
$ DELETE/CHARACTERISTIC EAST
```

You must specify the characteristic name, not the number, in the DELETE command.

You must remove all references to the characteristic before you can delete the characteristic. If the system displays the following message, it means that a queue or job is referring to the deleted characteristic:

```
%DELETE-E-NOTDELETED, error deleting EAST  
-JBC-E-REFERENCED, existing references prevent deletion
```

Creating Batch Queues

Batch Processing

Batch processing allows a user to request that a command procedure execute noninteractively. A job running in this way runs in the background without any user intervention. The JOB_CONTROL process creates a detached process to run the requested job, and performs system maintenance functions for the batch process.



Note

The JOB_CONTROL process performs the following tasks for a batch process:

- Creates a batch process on behalf of the requesting user
 - Creates a log file in the user's directory
 - Executes SYS\$SYLOGIN command procedure
 - Executes the user's LOGIN.COM
 - Executes the requested job
 - When the job is complete, it:
 - Deletes the batch process
 - Queues the log file to SYS\$PRINT to be printed
 - Deletes the log file
-

Creating a Batch Queue

To create a batch queue, use the INITIALIZE command.

Format:

INITIALIZE/QUEUE/BATCH[/qualifiers] queue-name

The account used to enter this command must have OPER privilege.

Example: Creating a Batch Queue

The following example shows how to create a Batch Queue:

```
$ ❶INITIALIZE❷ /QUEUE❸ /BATCH❹/START ❺SYS$BATCH
```

Example Notes: Creating a Batch Queue

Notes on the Creating a Batch Queue example include:

- ❶ The INITIALIZE command is used to initialize disks, tapes, and queues.
- ❷ /QUEUE informs the INITIALIZE command to initialize a queue.
- ❸ /BATCH informs the INITIALIZE command that the type of queue is BATCH.
- ❹ The /START qualifier causes the queue to start.
- ❺ The default queue for the DCL SUBMIT command is SYS\$BATCH.

Creating Print Queues

Print Queues

Printing files is a function that most systems perform. Output devices cannot be shared simultaneously among users; the selected device has to complete one job before it starts another. You manage the synchronization of multiple print requests by using print queues.

To create a print queue, the system manager must:

- Set device characteristics for printers.
- Create output execution queues.
- Create generic queues.

Setting Device Characteristics for Printers

Use the SET PRINTER command to set device characteristics for printers.

Format:

SET PRINTER/device-characteristic printer-name[:]

Example: Setting up Device Characteristics

The following example uses SET PRINTER to set up device characteristics:

```
$ SET PRINTER①/PAGE=60②/WIDTH=80③/LOWERCASE④/TRUNCATE ⑤LPA0:
$ SET PRINTER⑥/LA11⑦/UPPERCASE⑧/WRAP ⑨LPB0:
```

Example Notes: Setting Up Device Characteristics

Notes on the Setting up Device Characteristics include:

- ① /PAGE establishes the length of the output page as 60 lines.
- ② /WIDTH establishes the width of the output page as 80 characters.
- ③ /LOWERCASE enables lowercase printing.
- ④ /TRUNCATE causes the printer to truncate data if a line exceeds the width specified with the /WIDTH qualifier.
- ⑤ The printer name is LPA0:.
- ⑥ /LA11 establishes it as an LA11 printer.
- ⑦ /UPPERCASE disables lowercase printing.
- ⑧ /WRAP causes the printer to perform a carriage return or line feed whenever the line width (as specified by the /WIDTH qualifier) is reached. The rest of the record will continue on the next line.
- ⑨ The printer name is LPB0:.

Creating an Output Execution Queue

To create an output (device) execution queue, use the DCL INITIALIZE command.

Format:

INITIALIZE/QUEUE/DEVICE [/qualifiers] queue-name

Creating Generic Print Queues

Generic print queues are created in the same way that generic batch queues are created. Use the /GENERIC qualifier to the INITIALIZE/QUEUE command.

The following command creates and starts a generic queue, SYS\$PRINT, which will feed any print execution queues that have generic printing enabled.

```
$ INITIALIZE/QUEUE/START/GENERIC SYS$PRINT
```

The following command creates and starts a generic queue, SYS\$PRINT, which will feed the execution queues LPA0 and LPB0.

```
$ INITIALIZE/QUEUE/START/GENERIC=(LPA0,LPB0) SYS$PRINT
```


Creating Additional Queue Managers

Additional Queue Managers

You can create additional queue managers to help distribute the workload. This distribution might be necessary due to CPU or memory shortages in an OpenVMS Cluster environment. You can distribute the queues among queue managers based on:

- Batch versus print queues
- Nodes on the OpenVMS Cluster system
- Other conditions unique to your site

If you use additional queue managers, you are responsible for assigning the queues to the correct queue manager. If not specified, the default queue manager is selected.

Creating Additional Queue Managers

You must add the new queue manager information to the queue database. Do this with the /ADD qualifier on the START/QUEUE/MANAGER command.

Format:

**START/QUEUE/MANAGER/ADD/NAME_OF_
MANAGER=name/ON=(node1,node2,...)**

This adds the name to the database and starts the new queue manager. It is important to specify the nodes that are to be used for the additional queue manager.



Note

When adding a new queue manager, do NOT use the /NEW_VERSION qualifier. New queue managers share the same database with the current set of queue managers.

Example: Creating Additional Queue Managers

The following example shows how to create an additional queue manager:

- ❶ `$ SHOW QUEUE/MANAGER/FULL`
Master file: SYS\$SYSROOT:[SYSEXE]QMAN\$MASTER.DAT;

Queue manager SYS\$QUEUE_MANAGER, running, on BARNUM::
/ON=(BARNUM)
Database location: SYS\$COMMON:[SYSEXE]
- ❷ `START/QUEUE/MANAGER/ADD/NAME_OF_MANAGER=OTHER_MGR -`
`_ $ /ON=(LIONS,TIGERS,*)`
- ❸ `SHOW QUEUE/MANAGER/FULL`
Master file: SYS\$SYSROOT:[SYSEXE]QMAN\$MASTER.DAT;

Queue manager SYS\$QUEUE_MANAGER, running, on BARNUM::
/ON=(BARNUM)
Database location: SYS\$COMMON:[SYSEXE]
- ❹ Queue manager OTHER_MGR, running, on LIONS::
/ON=(LIONS,TIGERS,*)
Database location: SYS\$COMMON:[SYSEXE]

Example Notes: Creating Additional Queue Managers

Notes on the Creating Additional Queue Managers include:

- ❶ Begin with the default queue manager, SYS\$QUEUE_MANAGER.
- ❷ Start the additional queue manager.
- ❸ Now there are two queue managers.
- ❹ Information on the new queue manager, OTHER_MGR.

Creating Queues Under Multiple Queue Managers

Queues Under Multiple Queue Managers

The queue manager is assigned when you create the queue. Do this with the `INITIALIZE/QUEUE` command.

Format:

`INITIALIZE/QUEUE/NAME_OF_MANAGER=name queue-name`

When creating queues under multiple queue managers, if you do not name the queue manager in the command, the system uses the default queue manager `SYSS$QUEUE_MANAGER`.



Note

To change the queue manager for an existing queue, you must delete and reinitialize the queue, specifying the new queue manager's name in the `INITIALIZE/QUEUE` command.

Example: Creating Queues Under Multiple Queue Managers

The following example shows how to create queues under multiple queue managers:

```
❶ $ INITIALIZE/QUEUE/BATCH/START/NAME_OF_MANAGER=OTHER_MGR BATCH1
❷ $ INITIALIZE/QUEUE/BATCH/START/NAME_OF_MANAGER=OTHER_MGR BATCH2
❸ $ INITIALIZE/QUEUE/BATCH/START/NAME_OF_MANAGER=OTHER_MGR -
    $ /GENERIC=(BATCH1, BATCH2) NEWBATCH
❹ $ INITIALIZE/QUEUE/GENERIC/START -
    $ /NAME_OF_MANAGER=OTHER_MGR NEW_PRINT
    $
```

Example Notes: Creating Queues Under Multiple Queue Managers

Notes on the Creating Queues Under Multiple Queue Managers include:

- ❶ Create a batch execution queue using OTHER_MGR.
- ❷ Create another batch execution queue using OTHER_MGR.
- ❸ Create a generic batch queue that feeds the two execution queues BATCH1 and BATCH2.
- ❹ Create a print queue using OTHER_MGR.



Note

If the user uses the SUBMIT/PRINTER command, be sure to check that the batch queue and the print queue are BOTH being controlled by the same queue manager. Under OpenVMS Version 6.0, the SUBMIT/PRINTER command will not check this for multiple queue managers.

To find out which queue manager is controlling which queue, the user will need to either ask the system manager or use the F\$GETQUI lexical to extract the information.

Setting Up Logical Ports

Logical Ports

If your system is part of a network, and you will be accessing queues on other nodes, you must establish logical ports on the LAT.

Setting Up Logical Ports for LAT Printers

To set up a printer attached to a LAT port:

- Create a logical LAT port on each OpenVMS service node.
- Map the logical port with a physical port or service on the terminal server.

The file `SY$MANAGER:LAT$SYSTARTUP.COM` contains a template for this procedure.

Example: Setting Device Characteristics for a Logical Port

The `SET TERMINAL` command sets device characteristics for an output device attached to a terminal port.

```
❶ $ SET TERMINAL LTA3332: /DEVICE=LA210 /WIDTH=255 /PAGE=66 -  
   _$ /NOBROADCAST/PERMANENT  
❷ $ SET TERMINAL LTA3331: /SPEED=9600/PAGE=100/WIDTH=200 -  
   _$ /DEVICE=LN03 /NOBROADCAST /NOECHO /HARDCOPY -  
   _$ /NOTYPE_AHEAD /PERMANENT
```

Example Notes: Setting Device Characteristics for a Logical Port

Notes on Setting Device Characteristics for a Logical Port include:

- ❶ Set characteristics for the LA210 printer attached to LAT port LTA3332.
- ❷ Set up the LN03 printer attached to LAT port LTA3331.

Setting Up Spooled Devices

Spooling

Certain application programs print output by writing data directly to an output device rather than submitting it to a queue.

Spooling allows application programs to write output to an intermediate storage device so that the printer targeted by the application remains available to other users while the program is executing.

Creating a Spooled Output Device

To spool a device, use the SET DEVICE command. This command requires OPER privilege.

Format:

SET DEVICE/SPOOLED[=(*queue-name*[:],*int-disk*[:])] *device*

- The *queue-name* is optional. If you do not specify it, the default is a queue with the same name as the device.
- If you do not specify the intermediate disk (*int-disk*), the default is the system disk. The intermediate disk must have sufficient space, and if quotas are enabled on it, users must all have a quota entry. If specified, this disk must be mounted before setting it spooled.
- The *device* specifies the physical device to be spooled. This parameter is required.

Example: Setting Up Spooled Output devices

The following example shows how to set up spooled output devices using SET DEVICE:

- ❶ \$ SET DEVICE/SPOOLED=(SYS\$PRINT, SYS\$SYSDEVICE_ LPA0:
- ❷ \$ SET DEVICE/SPOOLED=(MKTG\$LN03_1, SYS\$DEVICE) LTA3331:
- ❸ \$ SET DEVICE/SPOOLED=(LA210\$PRINT, SYS\$SYSDEVICE) LTA3332:

Example Notes: Setting Up Spooled Output Devices

Notes the Setting Up Spooled Output Devices example include:

- ❶ Spool the output device LPA0:, associating it with the storage device SYS\$SYSDEVICE and the queue SYS\$PRINT.
- ❷ Spool the LN03 device on LAT port LTA3331:, associating it with the storage device SYS\$SYSDEVICE and the queue MKTG\$LN03_1.
- ❸ Spool the LA210 device on LAT port LTA3332:, associating it with the storage device SYS\$SYSDEVICE and the queue LA210\$PRINT.

Changing Device Characteristics on a Spooled Device

To change characteristics on a spooled device, you must first unspool the device.

Perform the following tasks if a device characteristic needs to be changed on a spooled device:

1. Stop the associated queue using STOP/QUEUE/NEXT.
2. Unspool the device using SET DEVICE/NOSPOOL.
3. Make changes using the SET commands.
4. Set the device spooled using SET DEVICE/SPOOL.
5. Start the queue using START/QUEUE.

Controlling Batch and Print Jobs

Overview

To control system resources efficiently, the system manager needs to control how batch and print jobs execute.

Topics

In this section the following topics are discussed:

- Controlling batch queue execution
- Using device control libraries
- Saving and restoring the queue database
- Using command procedures to automate queue creation
- Using forms

Controlling Batch Queue Execution

For performance reasons, and to enable interactive users to efficiently use system resources on a busy system, the system manager may have to control how batch queues execute.

To do this, the system manager can:

- Control the number of concurrent jobs executing in batch.
- Establish a scheduling priority for system jobs.
- Define batch working set limits.

Limiting the Number of Jobs

Use the `/JOB_LIMIT` qualifier to the `INITIALIZE` command to specify a limit for concurrently executing jobs.

The following command sets the job limit to 2.

```
$ INITIALIZE/QUEUE/BATCH/START/JOB_LIMIT=2 SYS$BATCH
```

- If there are more than two jobs in the queue, other jobs will be `PENDING`, waiting for a job to complete.
- As jobs complete, `PENDING` jobs will be moved into the queue for processing.

Establishing the Scheduling Priority for Batch Jobs

Normally, interactive processes should be favored to use the system, and batch jobs should run in the background. To ensure that this happens, the system manager can initialize a batch queue with a base scheduling priority lower than that of interactive processes (4).

Use the `/BASE_PRIORITY` qualifier to the `INITIALIZE` command to establish the base scheduling priority.

The following command establishes a base scheduling priority of 2 for all jobs executed by this queue.

```
$ INITIALIZE/QUEUE/BATCH/START/BASE_PRIORITY=2 SYS$BATCH
```

Controlling Batch Process Working Sets

Each process on a system consumes some amount of physical memory known as its working set.

Use the /WSEXTENT qualifier to limit the amount of physical memory for batch jobs.

The following command establishes a maximum working set value of 10,000 pages of physical memory.

```
$ INITIALIZE/QUEUE/BATCH/START/WSEXTENT=10000 SYS$BATCH
```

If not specified, each batch process takes on the user's default working set values. This could result in some users using more memory than others.

Using Device Control Libraries

Device Control Libraries

A **device control library** is a text library that contains user-written modules. These modules direct printers to perform certain functions before, during, and/or after the printing of a requested job.

You can use device control library modules to insert device-dependent escape sequences that set up a programmable printer for selected print options. These options include point size, character set, and bold or italic print. You can also use these modules to insert text at specific points in the processing of a print job on any printer.

Types of Device Control Modules

There are three types of device control modules.

This module type...	Is placed:
SETUP	At the beginning of a file
PAGE_SETUP	At the beginning of each page
RESET	At the end of each job

Steps for Using Device Control Libraries

To use a device control library module, perform the following steps:

Step	Task
1	Create the device control library.
2	Determine the contents of the module.
3	Create the module file.
4	Insert the module into the library.
5	Assign the device control library to a queue.

Creating a Device Control Library

To create a device control library, use the `LIBRARY/CREATE` command.

Format:

LIBRARY/CREATE/TEXT SYS\$LIBRARY:library-name.TLB

Determining the Contents of a Device Control Library

The system manager must select either the text to be inserted in the printed file or the escape sequences needed for the desired printer setup.

To determine the proper escape sequences for a printer option, refer to the operation guide for the specific printer.

Creating a Device Control Module

Create and edit a module file as you would any other text file. Enter the appropriate escape sequences, carriage control characters, or text.

Ensure that the text editor you use does not insert a carriage return or line feed at the end of your file. This will affect your printer output.

The suggested file type for a device control module is TXT.

Inserting a Module into a Device Control Library

Use the LIBRARY/INSERT command to insert a module into a device control library.

Format:

LIBRARY/INSERT/TEXT library-spec module-spec

Assigning a Library to a Queue

To assign a device control library to an output queue, specify the /LIBRARY qualifier with the INITIALIZE/QUEUE or START/QUEUE command.

Format:

INITIALIZE/QUEUE/START/LIBRARY=library-spec queue

START/QUEUE/START/LIBRARY=library-spec queue

- The value of *library-spec* specifies the device control library.
- The *queue* is the name of the output queue.

Listing the Modules in a Library

Use the LIBRARY/LIST command format to display a listing of all modules contained within a specified library.

Format:

LIBRARY/LIST SYSS\$LIBRARY:library-name.TLB

User Access to Device Control Modules

Users can access device control modules in two ways:

PRINT/SETUP=module	This command requests that a specific module be used by the PRINT command. If the module is misspelled or incorrect in some way, DCL does <i>not</i> detect an error. Validity is not checked until the job attempts to print. The job will not print and the user is not notified unless the /NOTIFY qualifier is used in the print request.
PRINT/FORM=form_name (or) PRINT/FORM=form_number	This command can be used if the system manager has associated a module with a particular form. This method is recommended, as it prevents users from losing print jobs.

PRINT/FORM is the preferred method.

Creating Forms for Setup and Page Setup Modules

To associate device control modules with a form, use the /SETUP=module or /PAGE_SETUP=module qualifiers with the DEFINE/FORM command.

Use this qualifier...	To perform this function:
/SETUP	Sends information to the printer when the form is mounted, before each file of a job is printed.
/PAGE_SETUP	Sends information to the printer before each page of a job.

Assigning a Reset Module to a Queue

If you need to use multiple modules with printers, you should reset the printer to a default setting after each print job so that users with standard print requests can use the printer without using a /FORM qualifier.

To assign a module to an output execution queue to reset the printer to a known state at the end of each job, use the /SEPARATE=RESET=module qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command.

Format:

INITIALIZE/QUEUE/SEPARATE=RESET=list queue

Where *list* is a list of one or more modules to be used upon reset.

The reset sequence contained in the module(s) resets the printer at the end of each job. It also resets the printer when the queue is started to ensure that the first job prints correctly.

Because the /SEPARATE qualifier specifies mandatory queue options, the RESET module you specify is sent to the queue at the end of every job. The user cannot override this option.

Example: Using Device Control Libraries

The following example creates a device control library and associates a queue with this library.

- ❶ \$ LIBRARY/CREATE/TEXT SYS\$LIBRARY:MYDEVCTL.TLB
\$ EDIT MODULE1.TXT
!enter printer escape sequences or text for module1
- ❷ \$ EDIT MODULE2.TXT
!enter printer escape sequences or text for module2
- ❸ \$ LIBRARY/INSERT SYS\$LIBRARY:MYDEVCTL.TLB/TEXT MODULE1
\$ LIBRARY/INSERT SYS\$LIBRARY:MYDEVCTL.TLB/TEXT MODULE2
- ❹ \$ INITIALIZE/QUEUE/START/ON=TTA9:/LIBRARY=MYDEVCTL PDQ_QUE
\$ SET QUEUE/SEPARATE=RESET=MODULE2 PDQ_QUE
- ❺ \$ SHOW QUEUE/FULL PDQ_QUE
Terminal queue PDQ_QUE, idle on T0AD::TTA9, mounted form DEFAULT
/BASE_PRIORITY=4 /DEFAULT=(FEED,FORM=DEFAULT) /LIBRARY=MYDEVCTL
/OWNER=[SYSTEM] /PROTECTION=(S:M,O:D,G:R,W:S)/SEPARATE=(RESET=(MODULE2))
- ❻ \$ DEFINE/FORM/SETUP=MODULE1/STOCK=DEFAULT FORM1 1
- ❼ \$ PRINT/FORM=FORM1 REPORT.TXT, MEMO.TXT/QUEUE=PDQ_QUE
Job REPORT (Queue PDQ_QUE, entry 619) started on PDQ_QUE

Example Notes: Using Device Control Libraries

Notes on the Using Device Control Libraries example include:

- ❶ Create a new library.
- ❷ Create the modules.
- ❸ Insert the modules into the library.
- ❹ Initialize the queue, specifying the library to use and the module to use for reset.
- ❺ Display the queue.
- ❻ Create a form specifying a setup module.
- ❼ Print a file using the newly created modules in the library.

Saving and Restoring the Queue Database

Saving the Queue Database

Once all the queues have been created and the forms and characteristics defined, the system manager should save the queue database so that it can be restored when needed. This will prevent having to duplicate the work necessary to establish all queues on a system.

Steps for Saving the Queue Database

Once all the forms, characteristics, and queues have been created, the system manager should make a copy of the databases.

The following procedure saves a copy of the queue database while the queuing system is functioning.

Step	Task	Comments and Commands
1	Convert the master file.	Use the Convert utility to extract a copy of the master file. <i>Backup-master-file</i> is the name of the file that will contain the copy of the master file.
	CONVERT/SHARE QMAN\$MASTER.DAT backup-master-file	
2	Convert the queue file.	Use the Convert utility to extract a copy of the queue file. File is the name of the file that will contain a copy of the queue file.
	CONVERT/SHARE SYSS\$QUEUE_MANAGER.QMAN\$QUEUES file	
3	Save the files.	Use the Backup utility to save the files to either removable media or another disk location.
	BACKUP/LOG input-specifier output-specifier	

Example: Saving the Queue Database

The following example saves a copy of the queue database. If the system contained multiple queue managers, the following conversion steps would be repeated for each queue database.

```
$ SET DEFAULT SYS$COMMON:[SYSEXE]
❶ $ CONVERT/SHARE QMAN$MASTER.DAT MASTERFILE_9SEP.KEEP;
❷ $ CONVERT/SHARE SYS$QUEUE_MANAGER.QMAN$QUEUES QFILE_9SEP.KEEP;
❸ $ INITIALIZE MUA0: QDB
    $ MOUNT/FOREIGN MUA0:
    %MOUNT-I-MOUNTED, QDB mounted on _LILITH$MUA0:
❹ $ BACKUP/LOG MASTERFILE_9SEP.KEEP,QFILE_9SEP.KEEP -
    _$ MUA0:QDB_9SEP.SAV/LABEL=QDB
    %BACKUP-S-COPIED, copied SYS$COMMON:[SYSEXE]MASTERFILE_9SEP.KEEP;
    %BACKUP-S-COPIED, copied SYS$COMMON:[SYSEXE]QFILE_9SEP.KEEP;
❺ $ DISMOUNT MUA0:
```

Example Notes: Saving the Queue Database

Notes on the Saving the Queue Database example include:

- ❶ Use the Convert utility to extract a copy of the master file.
- ❷ Use the Convert utility to extract a copy of the queue file.
- ❸ Initialize the tape.
- ❹ Use the Backup utility to save a copy of the files on the tape.
- ❺ Dismount the tape.

Restoring the Queue Database

If the queue database has been saved, it can be restored when necessary.

- All queues, forms, characteristics, and queue manager information will be restored.
- Information about jobs in the queues will not be restored.

Steps for Restoring the Queue Database

To restore previously saved queue database files, perform the following steps. If the system contained multiple queue managers, the following restoration steps would be repeated for each queue database.

Step	Task
1	Ensure that the appropriate queue manager is not running. If the queue manager is running, stop it by entering the DCL command STOP/QUEUE/MANAGER/CLUSTER[/NAME_OF_MANAGER].
2	Delete all three queue database files for the specific queue manager. You must delete all three files, even if only one or two of them has been lost.
3	Restore the queue database files for the specific queue manager. Use the Backup utility to restore the appropriate master file and queue file from the save set that you created. If the master file or queue file is stored in a location other than the default, ensure that you restore them to the correct location, or specify the new location when starting the queue manager. Always restore both the master file and the queue file of the database, even if only one of those files was lost.
4	Start the necessary queue manager. Use the START/QUEUE/MANAGER[/NAME_OF_MANAGER] command as you would normally. Do not enter the /NEW_VERSION qualifier. A new (empty) journal file is automatically created.

Example: Restoring the Queue Database

The following example restores the queue database files. If the system contained multiple queue managers, the following restoration steps would be repeated for each queue database.

- ❶ STOP/QUEUE/MANAGER/CLUSTER
SET DEFAULT SYS\$COMMON:[SYSEXEXE]
- ❷ DELETE SYS\$QUEUE_MANAGER.QMAN\$JOURNAL; , SYS\$QUEUE_MANAGER.QMAN\$QUEUES; , -
_ \$ QMAN\$MASTER.DAT;
\$ MOUNT/FOREIGN MUA0:
%MOUNT-I-MOUNTED, QDB mounted on _LILITH\$MUA0:
- ❸ BACKUP/LOG MUA0:QDB_9SEP.SAV/SELECT=[SYSEXEXE]MASTERFILE_9SEP.KEEP; -
_ \$ QMAN\$MASTER.DAT;
%BACKUP-S-CREATED, created SYS\$COMMON:[SYSEXEXE]QMAN\$MASTER.DAT;1
\$ SET MAGTAPE/REWIND MUA0:
- ❹ BACKUP/LOG MUA0:QDB_9SEP.SAV/SELECT=[SYSEXEXE]QFILE_9SEP.KEEP; -
_ \$ SYS\$QUEUE_MANAGER.QMAN\$QUEUES
%BACKUP-S-CREATED, created SYS\$COMMON:[SYSEXEXE]SYS\$QUEUE_MANAGER.QMAN\$QUEUES;1
\$ DISMOUNT MUA0:
- ❺ START/QUEUE/MANAGER

Example Notes: Restoring the Queue Database

Notes on the Restoring the Queue Database example include:

- ❶ Stop the queue manager.
- ❷ Delete the queue database files.
- ❸ Restore the master file.
- ❹ Restore the queue file.
- ❺ Start the queue manager.



Note

The OpenVMS operating system does not include software to support PostScript printing.

To print PostScript files, you must have either of the following:

- A printer capable of printing PostScript files and the supporting software.
- Software that provides PostScript to sixel printing and a supported printer.

The OpenVMS batch and print queuing system enables users to print files on output devices attached to the local system or OpenVMS Cluster.

The Distributed Queuing Service (DQS) is a layered product that extends the printing capabilities of the OpenVMS queuing system to a distributed environment. DQS enables users to print files on output devices attached to remote OpenVMS nodes in your network.

Using Command Procedures to Automate Queue Creation

Automating Queue Creation

Many of the tasks associated with defining and starting queues can be stored in command procedures that can be run each time the system is started. This prevents the system manager from having to redo all the work necessary to start batch and print queues whenever the system must be restarted.

Device Characteristics

Once you have set up the device characteristics for the printers on your systems, you can put the commands used into a command procedure that can be run each time the system boots.

Use the SET HOST/LOG command to create an interactive log of commands. You can use this log file to create the command procedure.

Spooling Devices

Once you have set up spooled devices for selected printers on your systems, you can put the commands used into a command procedure that can be run each time the system boots.

Use the SET HOST/LOG command to create an interactive log of commands that can be used to create the command procedure.

Automatic Startup at System Initialization

The autostart feature of queues simplifies the startup procedures during system initialization. Perform one of the following:

- Place the following commands in the site-specific startup procedure `SYSS$MANAGER:SYSTARTUP_VMS.COM`.
(or)
- Place the commands in a separate command procedure and invoke it from `SYSS$MANAGER:SYSTARTUP_VMS.COM`.

The following commands should be used, in order, every time the system is initialized.

Step	Task
1	Execute commands to set device characteristics.
2	Execute commands to spool devices.
3	Execute commands to start the nonautostart queues.
4	Enter the ENABLE AUTOSTART/QUEUE command to start all autostart queues.

**Note**

- If the queue manager was running before shutdown, it will automatically be started at boot time.
 - Once forms and characteristics are created, the definitions are stored in the queue database.
 - If generic queues are not stopped during shutdown, they do not need to be started at boot time.
-

Generic Queue Startup

Generic queues are not automatically stopped when the system shuts down. Thus, they do not need to be started at system boot time. No commands are necessary in the site-specific startup command file for generic queues.

Using Forms

Forms

Every output execution queue has an associated form. Each form has an associated paper stock. The stock of the requested form must match the stock of the printer, otherwise, the job will not be scheduled until they match.

Forms are used in the following ways:

- Control page width, length, and margin size.
- Control line overflow by truncating or wrapping lines.
- Specify device control library modules to set up the device at the start of each page, each file, or both.
- Pause output at the end of each page so that a new sheet of paper can be inserted. (Use this for printers that require sheet feed paper.)
- Specify that a certain paper stock must be physically loaded in the printer.

Managing Forms

To manage forms, a system manager must be able to:

- Define forms
- Display forms information
- Delete forms

If your printing needs are limited, you do not need to use special forms. Compaq supplies a systemwide default form (named DEFAULT) for all queues on an OpenVMS operating system.

Defining a Form

To create a form, use the DEFINE/FORM command.

Format:

DEFINE/FORM form-name form-number [/qualifiers]

Specifying a Form

You can specify any of the following qualifiers with the DEFINE/FORM command.

To specify this characteristic...	Use this qualifier:
The physical width of the paper in characters.	/WIDTH=n
The physical length of a form page in lines.	/LENGTH=n
Whether characters exceeding the line length specified by /WIDTH and /MARGIN should be discarded.	/[NO]TRUNCATE
Whether characters exceeding the line length specified by /WIDTH and /MARGIN should be wrapped to the next line.	/[NO]WRAP
The number of blank spaces for one or more of the four margin options: BOTTOM, LEFT, RIGHT, and TOP.	/MARGIN=(option=n[,...])
One or more device control modules that set up a device at the start of each page.	/[NO]PAGE_SETUP=(module[,...])
One or more device control modules that set up the device at the start of each file.	/SETUP=(module[,...])
Whether print jobs pause at the end of every physical page so that a new sheet of paper can be inserted.	/[NO]SHEET FEED
The type of paper stock to be associated with the form.	/STOCK=string
The string used to provide information about the form.	/DESCRIPTION=string

Example: Defining a Form

The following example defines a form, a stock, formatting options, and a text string.

```
$ DEFINE❶ /FORM MEMO 3❷ /STOCK=DEFAULT -  
_ $ ❸ /MARGIN=(TOP=2,BOTTOM=2,LEFT=6)/WIDTH=80/LENGTH=66/TRUNCATE -  
_ $❹ /DESCRIPTION="LN03 indented memo format"
```

Example Notes: Defining a Form

Notes on the Defining a Form example include:

- ❶ Define the form MEMO as the number 3.
- ❷ Define the stock to use as DEFAULT (the systemwide default).
- ❸ Define formatting options for the form.
- ❹ Define the text that will be displayed regarding this queue with the SHOW QUEUE/FORM/FULL command.

Providing Formatting Options Only

If you are creating forms only to provide different formatting options (and not to specify paper stock), specify the same stock type for each of the forms. That way, jobs requesting any of these forms will print on the same queue without requiring you to enter any additional commands to modify the queue.

Unless you specify the /STOCK qualifier, the form's stock is the same as the name of the form.

Example: Displaying Form Information

To display forms defined on a system, use the SHOW QUEUE/FORM/FULL command.

```
$ SHOW QUEUE/FORM/FULL
```

Form name	Number	Description
-----	-----	-----
DEFAULT	0	System-defined default /LENGTH=66 /MARGIN=(BOTTOM=6) /STOCK=DEFAULT /TRUNCATE /WIDTH=132
MEMO (stock=DEFAULT)	3	LN03 indented memo format /LENGTH=66 /MARGIN=(TOP=2,BOTTOM=2,LEFT=6) /STOCK=DEFAULT /TRUNCATE /WIDTH=80

If you know the name of the form that you want to display, you can specify the form name as a parameter to the command.

Assigning a Default Form for a Queue

If a user does not specify the /FORM qualifier when submitting a job with the PRINT command, the job uses the default form for the execution queue on which the job is printed.

To assign a default form for an output execution queue, specify the /DEFAULT qualifier with the INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command.

Format:

INITIALIZE/QUEUE/DEFAULT=FORM=name-or-number

Example:

```
$ INITIALIZE/QUEUE/START/DEFAULT=FORM=MEMO/ON=BAY::TTA3: MY_PRINT
```

Example: Displaying a Default Form for a Queue

To find the default form for a queue, use the DCL SHOW QUEUE command.

```
$ SHOW QUEUE MY_PRINT/FULL
Printer queue MY_PRINT, idle, on BAY::TTA3:, ❶mounted form MEMO ❷(stock=DEFAULT)
/BASE_PRIORITY=4 ❸DEFAULT=(FEED, FORM=MEMO (stock=DEFAULT)) /OWNER=[SYSTEM]
/PROTECTION=(S:M, O:D, G:R, W:S)
```

Example Notes: Displaying a Default Form for a Queue

Notes on the Displaying Form for a queue example include:

- ❶ The mounted form is MEMO.
- ❷ The stock for the mounted form is DEFAULT.
- ❸ If no form is requested on a PRINT command, it takes on the /DEFAULT value.

If the stock of the form requested does not match the stock of the mounted form, the job will not be scheduled.

Mounted Forms

A queue's mounted form is the form of the current job or, if no job is processing, the form of the last job printed in a queue.

As long as the stock of a job's form matches the stock of the queue's mounted form, the job is processed using the options in the job's form. The mounted form changes automatically to the form of the job being processed in the queue.

Changing the Mounted Form

If a job is pending because the forms do not match, operator intervention is required.

Perform the following steps as needed:

- If the requested form requires a different kind of paper, physically load the new paper.
- Change the mounted form with the SET QUEUE command.

SET QUEUE/MOUNTED_FORM=form queue-name

This change will stay in effect until the next job is requested. If the form is the same, the job will print.

Deleting a Form

To delete a form, use the DELETE/FORM command.

Format:

DELETE/FORM form-name

You must specify the form *name* with the DELETE/FORM command (not the form *number*).

Example:

```
$ DELETE/FORM MEMO
```

If the system displays the following messages, a queue or job exists with a reference to the form:

```
%DELETE-E-NOTDELETED, error deleting MEMO
-JBC-E-REFERENCED, existing references prevent deletion
```

You must remove all references to the form before you can delete the form.

Summary

Concepts

Managing Queues and Spooled Devices

- Execution queues accept batch and print jobs for processing.
- Generic queues hold jobs until they can be transferred to an execution queue for processing.
- Deleting a queue will remove it from the running system, and will remove the entry for that queue from the queue database.
- The default name for any queue is the name of the physical device associated with it.
- To make use of generic and execution queues, you may have to name queues in such a way that the name indicates the usage or limitations of the queue.
- The system manager can define characteristics and assign them to queues.
- Batch processing allows a user to request that a command procedure execute noninteractively.
- To create a print queue, the system manager must:
 - Set device characteristics for printers.
 - Create output execution queues.
 - Create generic queues.
- Additional queue managers help distribute the work load on an OpenVMS Cluster.
- When initializing the queue, the system manager needs to specify the name of the queue manager. If not specified, the default queue manager, `SY$QUEUE_MANAGER` is used.
- If the system is part of a network, and queues will be accessed on other nodes, the system manager will need to establish logical ports on the LAT.
- Spooling allows application programs to write output to an intermediate storage device so that the printer targeted by the application remains available to other users while the program is executing.

Controlling Batch and Print Jobs

- Control batch queue execution by setting job limits and scheduling priorities.
- Use device control libraries to set certain printer functions for all selected jobs.
- The queue database should be saved so that all queue information is available from a backup device if the queue database should become damaged in any way.
- Use command procedures to automate queue creation in the event of a system failure.
- Use forms to accommodate user needs for special printing.

Commands

Managing Queues and Spooled Devices

INITIALIZE/QUEUE

Creates a queue and specifies its characteristics.

INITIALIZE/QUEUE/GENERIC

Creates a generic queue and specifies its characteristics.

DELETE/QUEUE

Deletes a queue.

INITIALIZE/QUEUE/START/ON=device-name: queue-name

Creates a queue on a device and assigns it a name.

START/QUEUE

Starts a queue and specifies one or more characteristics for processing jobs on the queue.

DEFINE/CHARACTERISTIC

Creates a characteristic and assigns a name and number.

SHOW QUEUE/CHARACTERISTIC

Displays queue characteristics.

SET QUEUE

Specifies one or more characteristics for a queue.

DELETE/CHARACTERISTIC

Deletes a characteristic.

INITIALIZE/QUEUE/BATCH

Creates a batch queue and specifies its characteristics.

SET PRINTER

Sets printer characteristics.

**START/QUEUE/MANAGER/ADD/NAME_OF_
MANAGER=new_name/ON=(node1,node2,...)**

Starts an additional queue manager.

INITIALIZE/QUEUE/NAME_OF_MANAGER=name queue-name

Creates a queue using the specified queue manager.

SET TERMINAL

Sets device characteristics for an output device attached to a terminal port.

SET DEVICE/SPOOLED

Sets up a spooled device.

Controlling Batch and Print Jobs

INITIALIZE/QUEUE/BATCH/qualifier(s)

Qualifiers can be used to:

- Control the number of concurrent jobs executing in batch.
- Establish a scheduling priority for system jobs.
- Define batch working set limits.

DEFINE/FORM/qualifier(s)

Specifies one or more device control library modules that set up a device or a form.

CONVERT/SHARE QMAN\$MASTER.DAT backup-master-file

Extracts a copy of the master queue file.

CONVERT/SHARE SYS\$QUEUE_MANAGER.QMAN\$QUEUES file

Extracts a copy of the queue file.

BACKUP/LOG input-specifier output-specifier

Saves the files to either removable media or another disk location.

INITIALIZE/QUEUE/qualifier(s)

START/QUEUE/qualifier(s)

Initializes or starts a queue, specifying various characteristics.

PRINT/FORM=form-name

Requests that a print job be processed using the specified form.

DEFINE/FORM

Creates a form and assigns it a name and number.

DELETE/FORM

Deletes a form.

Managing LAT Services

Module 3

About This Lesson

Introduction

A LAT (local area transport) network is a collection of nodes connected by Ethernet or fiber distributed data interconnect (FDDI) for the purpose of sharing resources. These resources may include terminals, printers, and other devices.

This lesson presents the concepts necessary for managing LAT services.

Objectives

To manage LAT services, you should be able to:

- Describe basic LAT concepts.
- List the uses of LATCP.
- Create a LAT service alias.
- Create a remote print queue.

LAT Network Concepts

Overview

A LAT network is a collection of nodes connected by Ethernet or FDDI for the purpose of sharing resources.

A terminal server is a component on a local area network (LAN) that connects devices such as terminals, printers, modems and computers to the network. It allows these devices to be located anywhere in the LAN.

Topics

In this section the following topics are discussed:

- LAT network
- Terminal servers

Local Area Transport

LAT Network

The primary purpose of a LAT network is to provide access to computing resources.

A LAT network consists of:

- Service nodes — Computers on the LAN that offer services to users.
- Terminal server nodes — Communications servers that connect terminals and other devices to the LAN.

LAT Protocol

A **protocol** is a series of communications rules. A protocol is implemented with hardware and software.

The LAT network protocol is used by terminal servers to communicate on a LAN. LAT software and hardware implement the protocol on an OpenVMS system.

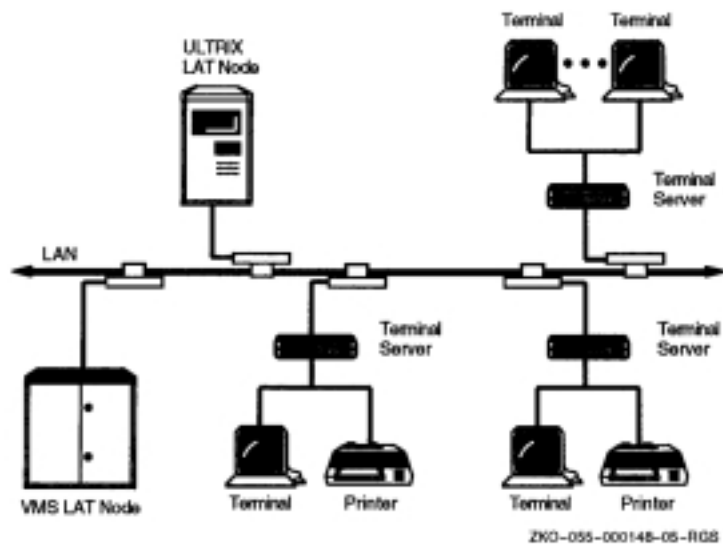


Note

LAT software is proprietary to Compaq Computer Corporation and is bundled with the OpenVMS operating system. (Bundled means the customer gets the product with the operating system license. Layered product licenses are purchased separately.)

Figure: LAT Network

The following figure shows an example of a LAT network configuration:



Service Nodes

A **service** is a name associated with an application or a port on a node. Every OpenVMS node on a LAT network offers its resources as a service. Each service is distinguished by a name assigned by the system manager.

- Most nodes offer general processing services (interactive login).
- A node can also be set up to offer a special application processing service or a limited service (Alpha only).

For example, an OpenVMS service node might offer three services:

- General processing
- Data entry application
- Stock quote application

Session

A **session** is a connection to a service.

A session is established when a user logs in to a system by using a terminal server.

Service Advertisements

An OpenVMS service node advertises its services over the network at regular intervals so that terminal servers know what network resources are available. The service announcement provides:

- Physical node name
- Service names
- Description of services

Terminal servers listen to LAN advertisements and record information.

When a user terminal requests service, the terminal server node requests connection to the appropriate OpenVMS service node.



Note

Use the LATCP SET NODE command with the /MULTICAST_TIMER qualifier to determine how often a service advertises itself to the LAT. Specify the interval in seconds.

Types of Service Requests

There are two types of service request.

- Local access — A user-initiated request for a connection from the user on a server port to a LAT service.

Example: Request from a user to connect to a VAX or Alpha service node.

- Remote access — A host-initiated request for a connection from a LAT service node to a terminal server.

Example: Request from a VAX or Alpha service node to print a job on a printer attached to a terminal server.

Terminal Servers

What Is a Terminal Server?

A **terminal server** is a LAN communications device that can establish and manage virtual circuit connections between serial devices and LAT hosts.

Most Compaq terminal servers are nodes without disks. When the terminal server is started up following a shutdown or power loss, the software image is downline loaded from another node on the network known as the load host.

All Compaq terminal servers support terminals and printers. Devices attach to the server at terminal server **ports**.

Terminal Server Operating Modes

Terminal servers operate in two modes.

- In **local mode**, an interactive user enters server commands to communicate directly with the server.
- In **service mode**, an interactive user communicates with a service. The server does not interpret data entered by the user. Data is passed to the service node.



Note

Local mode is distinguished by the prompt that appears on the port user's terminal. The default prompt is:

Local>

At the local prompt, the user can enter commands such as the following, which display the user's port characteristics on the terminal.

Local> SHOW PORT CHARACTERISTICS

To the user, the connection appears to be direct to the service node. The user sees the prompt for the operating system or application program, not the server's prompt.

Permanent Terminal Server Database

Values set in the permanent terminal server database remain constant through initializations, power losses and port logins.

- Privileged users can modify server and service characteristics in the permanent database using the DEFINE command in local mode. These changes do not take effect until the server is reinitialized.
- Nonprivileged users can modify some port characteristics using the DEFINE command in local mode. Changes to port characteristics become operational when the user logs out of the port and then logs back in or when the server is reinitialized.

Operational or Volatile Terminal Server Database

After initialization, the operational or volatile terminal server database is a copy of the permanent database.

- Privileged users can modify server and service characteristics with the SET command. Changes take effect immediately and remain in effect until the server is initialized.
- Nonprivileged users can modify some port characteristics in the operational database. Changes take effect immediately and remain in effect until the server is initialized or the port is logged out.



Note

Details of the databases are server specific. Students should refer to the manual for their particular terminal server.

Troubleshooting note: When a device works for a while, but then does not work after the system is rebooted, suspect that characteristics were set in the volatile database rather than in the permanent database.

Displaying Server Characteristics

A general user can enter commands in local mode at the terminal server prompt to display the information in the database.

```
Local> SHOW PORT CHARACTERISTICS
Local> SHOW SERVER CHARACTERISTICS
Local> SHOW SERVICE CHARACTERISTICS
```

Testing a Terminal Server Port

To verify that a terminal server port is working, log in to the server and test the port using the terminal server TEST PORT command.

Format:

TEST PORT COUNT *n*

Where the value of *n* indicates the number of test passes to be performed.

Example:

```
Local> TEST PORT COUNT 3
```

The terminal server sends out a rotating character pattern to the port, which is displayed on the terminal.

If the pattern is not consistent, there is something wrong with one or more of the following:

- Server
- Port
- Terminal
- Cabling out to the terminal



Note

When a remote print queue is not working, you can isolate the problem by placing a terminal at the end of the printer cable and conducting the port test. If the test is good, you can eliminate the problems listed above.

This does not eliminate the printer itself as the cause of the problem.

The next step is to reconnect the printer and try the COPY command. If COPY fails, the problem is with the printer.

Example: Testing a Terminal Server Port

In the following example, the test indicates that the port is working correctly because the test pattern is consistent.

```
Enter username> SYSTEM
```

```
Local> TEST PORT COUNT 3
```

```
!"#$%&'()*+,-./0123456789:;<literal>(<)=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[^_`abcd
```

```
"#$%&'()*+,-./0123456789:;<literal>(<)=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[^_`abcde
```

```
Local>
```

Using the LAT Control Program

Overview

The LAT Control Program (LATCP) is used to configure and control the LAT software from a LAT service node.

Topics

In this section the following topics are discussed:

- LAT Control Program



Note

You should create any necessary logical LAT links with LCP CREATE LINK before you turn on the LAT.

LAT Control Program

LATCP

The LAT Control Program (LATCP) is used to configure and control the LAT software from a LAT service node.

Functions of LATCP

The LAT Control Program allows the system manager to:

- Start and stop the LAT port driver (LTDRIVER on OpenVMS systems).
- Specify operational characteristics for your service node and its services.
- Display the status for your LAT service node.
- Show and zero LAT counters.

Running LATCP

To run LATCP, enter the following DCL command:

RUN SYS\$SYSTEM:LATCP

Example:

```
$ RUN SYS$SYSTEM:LATCP
LATCP>
```

To exit from LATCP, enter the EXIT command or press Ctrl/Z.

Sample LATCP Commands

To perform this function...	Use this LATCP command:
Start the LAT port driver on the local node.	SET NODE/STATE=ON
Stop the LAT port driver on the local node.	SET NODE/STATE=OFF
Display the specified LAT information.	SHOW (entity)
Modify the specified LAT information.	SET (entity)
Turn on network multicasts.	SET NODE
/ANNOUNCEMENTS	
Turn off network multicasts.	SET NODE
/NOANNOUNCEMENTS	

Setting Up a LAT Cluster Alias

Overview

In a clustered environment, it is possible to create a service name, called an alias, that equates to one of several cluster nodes. Users on the LAT can connect to the cluster alias. The actual node that they are connected to is determined by the LAT.

Topics

In this section the following topics are discussed:

- LAT cluster alias

LAT Cluster Alias

In a clustered environment, it is possible to create a service name, called an **alias**, that equates to one of several cluster nodes. Users on the LAT can connect to the cluster alias. The actual node that they are connected to is determined by a LAT algorithm, which selects the least loaded node. In this way, some load balancing is achieved.

Note that a LAT cluster alias is *not* the same as a DECnet cluster alias. Only terminals connected to the LAT can access the LAT cluster alias.

Creating a LAT Cluster Alias at System Startup

To create a LAT cluster alias, edit the site-specific startup file `SYSS$COMMON:SYSTARTUP_VMS.COM` and modify the line that invokes `SYSS$STARTUP:LAT$STARTUP.COM`.

1. Pass the service name as parameter P1.
2. Pass a P2 value of `/CONNECT=BOTH` to enable incoming and outgoing connections.

Example:

```
$ @SYSS$STARTUP:LAT$STARTUP "CLUSTER" "/CONNECT=BOTH"
```

Creating a LAT Cluster Alias from DCL

To create a service alias on individual nodes from the DCL prompt:

1. Log in to the appropriate node.
2. Set default to `SYSS$SYSTEM`.
3. Invoke LATCP and create the service on each node that should be part of the service.

Example: Creating a LAT Cluster Alias from DCL

The following example shows how to create a LAT cluster alias using DCL:

```
$ SET DEFAULT SYSS$SYSTEM
$ RUN LATCP
LATCP> CREATE SERVICE CLUSTER
LATCP> EXIT
```

Setting Up a Remote Print Queue

Overview

The system manager can create a remote print queue to allow users to access printers that are connected to the local area network through a terminal server.

Topics

In this section the following topics are discussed:

- Initial actions
- Step 1: Set up the terminal server
- Step 2: Set up the printer port on the server
- Step 3: Set up the application port
- Step 4: Set up the print queue

Initial Actions

Initial Actions

Before setting up a remote print queue:

- Verify the printer settings.
- Display the printer port characteristics.

Verifying Printer Settings

Obtain a test sheet from the printer and verify the device setting for:

- Character size
- Parity
- Flow control
- Baud rate

Example: Displaying Port Characteristics

Use the terminal server in local mode to display the characteristics of the port to which the printer is to be connected.

```
Local> SHOW PORT CHARACTERISTICS
```

Port 2:

Character Size:	8	Input Speed:	4800
Flow Control:	XON	Output Speed:	4800
Parity:	None	Modem Control:	Disabled

Access:	Remote	Local Switch:	None
Backwards Switch:	None	Name:	LN03_4
Break:	Disabled	Session Limit:	1
Forwards Switch:	None	Type:	Hard

Preferred Service: None

Authorized Groups: 0
(Current) Groups: 0

Enabled Characteristics:

DSRlogout, Inactivity Logout, Input Flow Control, Output Flow Control, Verification

Step 1: Set Up the Terminal Server

Steps for Setting Up the Terminal Server

To set up the terminal server:

1. Install and initialize the terminal server.
2. Select a terminal server name.
3. Set the terminal server name.
4. Verify the terminal server name.

Install and Initialize the Terminal Server

Follow the specific instructions for installing and initializing the terminal server.

Select a Terminal Server Name

Select a name for the terminal server. This name:

- Must be unique on the Ethernet
- Can be up to six characters long
- Must be written to both the permanent and volatile server database

Set the Terminal Server Name

Use the terminal server SET SERVER command to set the server name.

Format:

SET SERVER NAME server-name

For example, to set the server name to ZKO123:

```
Local> SET SERVER NAME ZKO123
```

Verify the Terminal Server Name

Use the terminal server SHOW SERVER command to verify the server name.

Format:

SHOW SERVER

Example: Verifying the Terminal Server Name

The following example uses the SHOW SERVER command to verify the terminal server name.

Local> SHOW SERVER

DECserver 500 V2.1.5 LAT V5.2 ROM V2.0.1 Uptime: 46 16:44:31

Address: 08-00-2B-08-C5-46 Name: ZK0123 Number: 0

.
.
.



Note

Because each terminal server has a different operating system, the commands to set up the terminal server vary from one product to another. The documentation for the terminal server to be used in the laboratory exercises should be available to the students.

The Terminal Server Configurator (TSC) utility is a software tool that resides on the load host and is used to modify the DECserver 500 permanent database. Because this is specific to the DECserver 500, it is not discussed here.

The DECserver 500 has three databases, not two

Step2: Set Up the Printer Port on the Server

Steps for Setting Up the Printer Port

To set up the printer port:

1. Give the port an appropriate name.

For example, to assign the name PORT_2 to port number 2:

```
Local> SET|DEFINE PORT 2 NAME PORT_2
```

The expression SET|DEFINE indicates that on some servers the command is SET and on others it is DEFINE.

2. Modify the following port characteristics:

```
ACCESS REMOTE
```

```
AUTOBAUD DISABLE
```

```
Local> SET|DEFINE PORT 2 ACCESS REMOTE AUTOBAUD DISABLE
```

3. Set the port to match the device speed, parity size, and flow control.

```
Local> SET|DEFINE PORT n SPEED baud-rate
```

4. Disable all enabled characteristics except for INPUT FLOW CONTROL and OUTPUT FLOW CONTROL.

5. Use the SHOW PORT command to verify the settings.

```
Local> SHOW PORT 2
```

6. Test the port to verify that you receive a correct test pattern.

```
Local> TEST PORT 2 COUNT 20 WIDTH 80
```

7. Verify the contents of the name fields on both the SHOW SERVER and SHOW PORT displays.

Example: Setting Up a Printer Port

The following example sets up server port 2 for a 4800 baud printer.

```
Local> SET PORT 2 NAME PORT_2
Local> SET PORT 2 ACCESS REMOTE
Local> SET PORT 2 AUTOBAUD DISABLED
Local> SET PORT 2 SPEED 4800
Local> SET PORT 2 PREFERRED NONE DEDICATED NONE
Local> SAVE PORT 2
Local> LOGOUT PORT 2
```

Use the SAVE PORT command only if the server has nonvolatile storage for parameters.



Note

To determine the port access characteristic as REMOTE and to disable the characteristic AUTOBAUD where n=port-number on the server, use the following command:

```
Local> SET|DEFINE PORT n ACCESS REMOTE AUTOBAUD DISABLE
```

These definitions are made permanent by being stored in the server permanent database. This happens on a low-end server when you log out of the port, and on a high-end server with the SAVE PORT command.

On the DECserver 500 series server, use the Terminal Server Configurator:

```
TSC>DEFINE PORT 2 SPEED 4800
```

Step 3: Set Up the Application Port

Steps for Setting Up the Application Port

An application port must be created and mapped on each node that will use the remote printer. This must be done every time that node is initialized.

1. Select an application port name.
2. Create the application port.
3. Map the application port to the terminal server port.
4. Verify the port mapping.
5. Make the port setup permanent.

Select an Application Port Name

Application ports are called LTAn, where n is a number from 1 to 9999.

Each application port name must be unique.

- Design a numbering system to keep the port names unique.
- Before assigning a port name, check to see if it exists.

Create the Application Port

Run LATCP and create the application port on each node in the OpenVMS Cluster that will access the printer.

```
LATCP> CREATE PORT LTAn/APPLICATION
```

The /APPLICATION qualifier specifies that a logical port on an OpenVMS node is to be an application port. It can be used to connect to a remote printer on a terminal server. Choices are /APPLICATION and /DEDICATED. The default is /APPLICATION.

**Note**

The LATCP CREATE PORT command creates a logical port on the local OpenVMS node that connects with a remote device on a terminal server. Alternatively, this command creates a logical port on your local OpenVMS node that connects with a specific service. The service can be offered by a terminal server or associated with one or more dedicated ports on a remote LAT service node.

Map the Application Port

To associate (map) the application port with the terminal server port, run LATCP on each node that will access the printer. Execute the SET PORT command.

Format:

SET PORT LTAn/APPLICATION/NODE=server/PORT=port

- The value of *n* specifies the port number.
- The *server* specifies the name of a terminal server to be logically associated with the specified application port.
- The *port* specifies the name of the remote port on a terminal server that supports a remote device.

The SET PORT command associates a logical port on the local OpenVMS node with a remote port on a terminal server that supports a device. Alternatively, it associates a logical port on the local OpenVMS node with a specific service. The service can be offered by a terminal server or associated with one or more dedicated ports on a remote LAT service node.

Verify the Port Mapping

Verify the mapping of the application port to the terminal server port.

In local mode on the terminal server, use the TEST PORT command.

Format:

TEST PORT port-number COUNT count-number

- Line printers use a COUNT of 10.
- Laser printers use a COUNT of 70.

On the node that is to access the printer, use the DCL COPY command.

Format:

COPY/LOG filename LTAn:

You can use the COPY command to test the printer because the printer has not been set spooled. After the device is set spooled, this command will not work.

Make the Application Port Setup Permanent

To make the application port setup permanent, add the commands to create and map the application port to the LAT startup command procedure.

- This command procedure, SYSS\$MANAGER:LAT\$SYSTARTUP.COM, is called from SYSTARTUP_VMS.COM.
- The call to invoke the LAT startup command procedure should take place after the DECnet software is up and running.

Example: Creating and Mapping an Application Port

The following example creates an application port named LTA8000 and maps the application port to the port PORT_2 on terminal server ZKO123.

```
LATCP> CREATE PORT LTA8000: /APPLICATION
LATCP> SET PORT LTA8000:/APPLICATION/NODE=ZKO123/PORT=PORT_2
```

Step 4: Set Up the Print Queue

Steps for Setting Up the Print Queue

Initialize the remote print queue using the same procedures as for regular print queues, with some important differences.

- Set up the printer characteristics.
- Set protection on the device to allow only the system to access the printer.
- Set the device spooled.
- Define forms.
(Optional step, execute only once on Version 5.5 and later systems.)
- Initialize the queue.
(Execute this step only once on Version 5.5 and later systems.)
- Start the queue.
- Test the queue.

Set Up the Printer Characteristics

Use the DCL SET TERMINAL command to establish remote printer characteristics. Include this command in a startup command procedure to set the printer characteristics on each node that will use the queue.

Format:

SET TERMINAL/qualifier(s) application-port-name

Example:

```
$ SET TERMINAL/PERMANENT/DEVICE=LN03/NOBROADCAST/PAGE=60 LTA8000:
```

To specify this characteristic...	Use this qualifier:
Terminal type	/DEVICE=terminal-type
Maximum number of characters per line (This value must be an integer in the range of 1 to 511, and should be set to the maximum number of characters for all print forms.)	/WIDTH=n
Number of print lines between form feeds (This value can be from 0 to 255. The default is 0, which treats a form feed as a line feed.)	/PAGE=n
Whether reception of broadcast messages is enabled (With a print queue, messages should not be printed in the middle of the output file.)	/NOBROADCAST
Speed of the device	/SPEED = baud-rate
Whether characteristics will survive terminal sessions	/PERMANENT



Note

Printer characteristics revert to their original values if the system is halted and restarted.

Set the Device Protection

The application port should only be accessed by the system and not by owner, group, or world.

Use the DCL SET PROTECTION/DEVICE command to set the device protection.

Format:

SET PROTECTION=(S:RWLP,O,G,W)/DEVICE LTAn:

To allow this access...	Assign this protection category:
Issue read requests to the device	R
Issue write requests to the device	W
Issue logical I/O requests to the device	L
Issue physical I/O requests to the device	P
No access	Null

Set the Device Spooled

Set the remote print queue device spooled to prevent applications from sending output directly to the printer.

Use the DCL SET DEVICE command to set the device spooled.

Format:

SET DEVICE LTAn:/SPOOLED



Note

Setting the device spooled prevents privileged users from sending a direct COPY command to the printer

Initialize and Start the Queue

Use the DCL INITIALIZE command to initialize and start the queue.

Format:

INITIALIZE/QUEUE/START/qualifier(s) queue-name

INITIALIZE qualifiers appropriate for print queues can be used. There are some qualifiers that are specific to remote print queues.

To specify this characteristic...	Use this qualifier:
Use the print symbiont designed specifically for use with the LAT protocol.	/PROCESSOR=LATSYM
Retain jobs that completed unsuccessfully. (Override any job retention option requested by a user for a job entered in the queue.)	/RETAIN=ERROR
Associate the queue with the application port, which was mapped previously to a particular server and server port.	/ON=[node::]LTAn[:]



Note

INITIALIZE/QUEUE/ON=

For output execution queues, you can include both the node name and the device name. By default, a queue executes on the same node from which you start the queue. The default device parameter is the same as the queue name.

Example: Initializing and Starting a Queue

The following command initializes and starts a remote print queue named LASER on application port LTA8000.

```
$ INITIALIZE/QUEUE/START/PROCESSOR=LATSYM/RETAIN=ERROR -  
_$ /DEFAULT=(NOBURST,NOTRAILER)/ON=LTA8000: LASER
```



Note

The DCL INITIALIZE/QUEUE/AUTOSTART_ON command designates an execution queue as an autostart queue and specifies the node, or node and device, on which the queue can be located. Both node and device must be specified for output queues.

Format:

/AUTOSTART_ON =(node::[device][,...])

When you enter the INITIALIZE/QUEUE command with the /AUTOSTART_ON qualifier, you must initially activate the queue for autostart, either by specifying the /START qualifier with the INITIALIZE/QUEUE command or by entering a START/QUEUE command. However, the queue will not begin processing jobs until you enter the ENABLE AUTOSTART/QUEUES command for a node on which the queue can run.

You cannot use this qualifier in conjunction with the /ON or /GENERIC qualifier. However, if you are reinitializing an existing queue, you can specify the /AUTOSTART_ON qualifier for a queue that was previously created or started with the /ON qualifier. Doing so overrides the /ON qualifier and makes the queue an autostart queue.

Test the Printer

Use the DCL PRINT command to test the printer setup by printing a file on the printer.

Format:

PRINT/QUEUE=remote-queue-name file-specification

Example:

```
$ PRINT/QUEUE=LASER MYFILE.TXT
```

Summary

Concepts

LAT Network Concepts

- The LAT (local area transport) protocol is used to provide terminals access to computing resources in a LAN.
- Terminal servers are devices that connect several terminals in a LAN.

Using the LAT Control Program

- The LAT Control Program (LATCP) is used to manage a LAT.

Setting Up a LAT Cluster Alias

- A LAT cluster alias allows LAT users to connect to a single service name that serves several machines in a cluster.

Setting Up a Remote Print Queue

- Before setting up a remote print queue:
 - Verify the printer settings.
 - Display the printer port characteristics.
- To set up the terminal server:
 - Install and initialize the terminal server.
 - Select a terminal server name.
 - Set the terminal server name.
- Verify the terminal server name.
- In local mode on the terminal server, use the commands appropriate to the specific server to set up the printer port.
- An application port must be created and mapped on each node that will use the remote printer. This must be done every time that node is initialized.

- Initialize the remote print queue using the same procedures as for regular print queues, with some important differences.
 - Set up the printer characteristics.
 - Set protection on the device to allow only the system to access the printer.
 - Set the device spooled.
 - Define forms.
 - Initialize the queue.
 - Start the queue.
 - Test the queue.

Commands

LAT Network Concepts

SHOW (entity)

Terminal server command to display information in the terminal server database.

TEST PORT

Terminal server command to verify that a terminal server port is working.

Using the LAT Control Program

SET (entity)

LATCP command to modify LAT information.

SHOW (entity)

LATCP command to display LAT information.

Setting Up a LAT Cluster Alias

CREATE SERVICE

LATCP command to create a service alias on an individual node.

Setting Up a Remote Print Queue

SET SERVER NAME

Terminal server command to establish a server name.

SHOW SERVER NAME

Terminal server command to verify a server name.

SET PORT

DEFINE PORT

Terminal server command to establish port characteristics.

TEST PORT

Terminal server command to verify the mapping of an application port to a terminal server port.

CREATE PORT

LATCP command to create an application port.

SET PORT

LATCP command to map an application port to a terminal server port.

SET TERMINAL

DCL command to establish remote printer characteristics.

SET PROTECTION

DCL command to establish protection for print queue device.

SET DEVICE/SPOOLED

DCL command to set remote print queue device to be spooled.

INITIALIZE/QUEUE/START

DCL command to initialize and start a remote print queue.

Managing Volumes

Module 4

About This Module

Introduction

Disk and tape volume management is a common task that system managers perform on a daily basis. This module presents the basic concepts of volume management.

Objectives

To manage disks and tapes on an OpenVMS system, you should be able to:

- Describe how devices are named in an OpenVMS system.
- Describe how OpenVMS Cluster systems support the SCSI as a storage interconnect.
- Describe the structure of the system disk directories.
- Obtain information about volumes.
- Prepare volumes for use.
- Handle common volume problems.

Review of OpenVMS Device Concepts

Overview

To perform operations with volumes on an OpenVMS system, the system manager must be able to name the device that contains the volume. Devices in an OpenVMS system are named according to a standard format. This section reviews the naming conventions for devices.

Topics

In this section the following topics are discussed:

- OpenVMS device names
- Devices in an OpenVMS Cluster system

OpenVMS Device Names

Device Names

Every device has a unique name in the format: **ddcu**

Dd	A two-letter device code.
c	A one-letter code that specifies the hardware controller for the device. (Controllers provide the interface between the bus and the device, or between two buses.)
u	The device unit number.

Parts of a Device Name

In a device name, the **device code** specifies the device type. The following table lists some common two-letter device codes.

This code...	Refers to this device:
CS	Console storage device
DU	RA80 or RA81 disk drive
MU	TK70 tape drive
LT	Terminal connected by means of a terminal server

For example:

- CSA1 = Console Storage, Controller A, Device Unit 1
- LTA251 = LAT, Controller A, Device Unit 251

Hardware Controller Number

The hardware controller number:

- Identifies the device controller
- Is represented by a letter from A to Z
- Is assigned by the system

Unit Number

The unit number:

- Indicates the position of the device on the controller
- Can be changed by:
 - Setting a button or switch on the device
 - Installing a unit plug on the device

Devices in an OpenVMS Cluster System

The MSCP (mass storage control protocol) server allows clusterwide sharing of devices that are connected locally to a VAX processor or to an intelligent, MSCP compliant controller.

When a device is MSCP served, any OpenVMS node in the cluster can mount and access that device as if it were a local device.

MSCP Server

MSCP served disks include:

- Disks local to CI members
- Disks on boot servers anywhere in the cluster
- Disks on disk servers anywhere in the cluster, including satellites
- HSC disks in mixed-interconnect clusters (when the MSCP server is running on one of the CI nodes)
- Integrated storage element (ISE) disks connected to a DSSI (Digital standard storage) interconnect in a mixed-interconnect cluster

The MSCP server decodes and services MSCP I/O requests sent by the disk class driver on remote cluster nodes.

Tape MSCP Server

When a tape device is MSCP served, any OpenVMS node in the cluster can mount and access that tape device as if it were a local device.

Like a local tape drive, an MSCP served tape drive can be used by only one process at a time.

**Note**

A common but erroneous statement is that the MSCP server allows a VAX system to emulate an HSC node. A correct statement is that the MSCP server software and the HSC and ISE are three different implementations of the MSCP protocol, on three different kinds of processors.

Because both DECnet and MSCP server software perform file access over the Ethernet, you may be asked to compare them.

- MSCP performs logical disk block I/O, and has no knowledge of the file system on the disk. DECnet FAL (File Access Listener) uses the file system and RMS for file access, therefore, it supports record-level access only.
- DECnet DFS (Distributed File Server) uses its own connection mechanism to eliminate many of the performance degradations of FAL. DFS performance over an Ethernet is very close to MSCP server performance over an Ethernet.
- MSCP depends on SCS, which supports Ethernet, DSSI, and CI for remote access. SCS performs no routing.

DECnet supports many communication media including CI, Ethernet, and synchronous and asynchronous links. DECnet can route I/O requests through multiple systems.

Students may ask if you can control MSCP parameters. This can be done by using CLUSTER_CONFIG.COM.

- MSCP_LOAD = 0 not to load, 1 to load
 - MSCP_SERVE_ALL = 0 not to serve disks, 1 to serve all disks (including HSC or ISE disks, so that satellites have access to them), 2 to serve only local disks.
-

Device Name Formats

Two formats are used in naming devices in a clustered environment:

- node\$device
- \$allocation-class\$device

Node\$device Format

Node\$device is used for devices that are directly connected to only one node.

- Zero is the default value for the ALLOCLASS SYSGEN parameter, forcing this format.
- Use **node\$device** to refer to devices on other nodes when specifying:
- A terminal on another node to OPCOM
- A printer on another node to the job controller

Allocation Class

Allocation class is a parameter set on a node that serves disks (an HSC or ISE controller or a VAX node running the MSCP server).

- Zero is the default.
- Setting the ALLOCLASS parameter to non-zero (1-255) enables the \$allocation-class\$device format for device names.
- Used for dual-pathed (including dual-ported and dual-hosted) devices, to provide a single name for the device.
- For any disk or tape device, all nodes that serve it to the cluster must have the same allocation class.

Duplicate Device Names

No two devices in a cluster can have the same name.

For example, if two nodes have allocation class 1, you cannot connect a drive named DUA0 to each node because there would then be two devices named \$1\$DUA0. To avoid this problem, change the unit number of one of the drives.

Naming Devices on a Local Node

When a device is on a local node:

- You can use its traditional name (for example, TXA2).
- You can prefix its name with the node name (for example, BARNUM\$TXA2).
- Each disk must have a unique volume label.



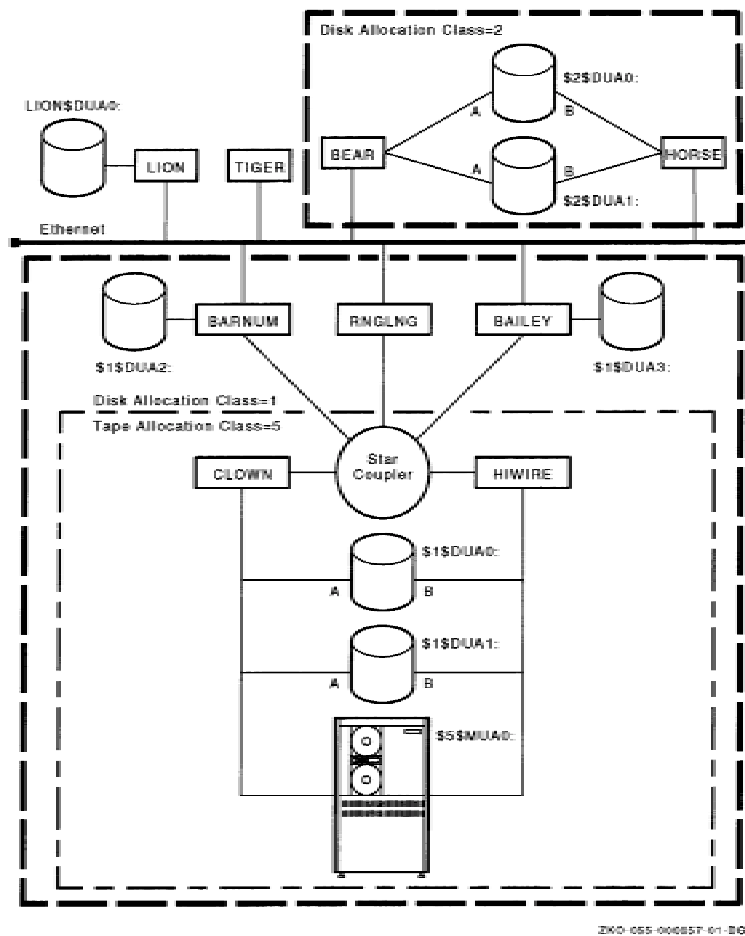
Note

The following terms are used in describing configurations.

This term...	Has this meaning:
Dual-ported	The disk has two cables coming out of the back. This means that the software/hardware can achieve better performance because of the interleaved data. This also allows two nodes to have direct access to the disk.
Dual-pathed	The disk has cables going to two different computers, so if one path fails, the data is still accessible by the other path. (You have to log in from the remaining node.)
Dual-hosted (or multiple hosted)	In a mixed-interconnect cluster, the disk is served to the satellite nodes by more than one CI node.

Figure: Sample OpenVMS Cluster System

This figure shows a sample cluster with allocation classes assigned to nodes. It also shows the device name of each disk and tape drive.



**Note**

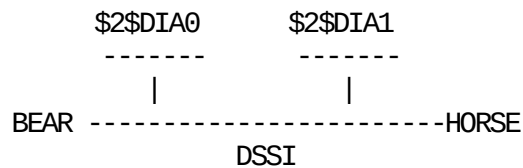
There are three disks in this figure that have the same unit number. (Example: DUA0: on LION.) If these disks are available clusterwide, the only difference in their names will be the node that serves them. (Example: BARNUM\$DUA0.)

All disks are dual ported in this example.

The figure shows only the nodes in the CIRCUS cluster. There might be other nodes on the same Ethernet, but they would not be able to mount these devices directly because they are not in the cluster.

You may want to discuss how dual-porting works.

Consider redrawing the figure with the disks between BEAR and HORSE as DSSI devices, named \$2\$DIA0 and \$2\$DIA1:



SCSI Clusters

Overview

One of the benefits of OpenVMS Cluster systems is that multiple computers can simultaneously access storage devices connected to an OpenVMS Cluster storage interconnect. Together, these systems provide high performance and highly available access to storage.

Topics

In this section the following topics are discussed:

- Interconnect Concepts
- Add-On SCSI Adapters
- Internal SCSI Adapters

Interconnect Concepts

Accessing SCSI Storage

With OpenVMS Alpha Version 6.2 or later, you can connect multiple Alpha hosts to a single SCSI bus to share direct access to SCSI storage devices. Before, this type of connection was supported by using an embedded SCSI adapter, an optional external SCSI adapter, or a special-purpose RAID (redundant array of independent disks) controller. Only one host could be connected to a SCSI bus.

Note

- A SCSI interconnect is also called a SCSI bus.
 - Host-based RAID (including host-based shadowing) and the MSCP server are supported for shared SCSI storage devices.
 - Multiple VAX and Alpha hosts can directly access SCSI devices in any of these ways:
 - ♦ CI interconnect with HSJ or HSC controllers
 - ♦ Digital Storage Systems Interconnect (DSSI) with HSD controller
 - ♦ SCSI adapters directly connected to VAX or Alpha systems
 - Use the OpenVMS MSCP server to access SCSI devices indirectly.
 - Another interconnect, such as a LAN, is required for host-to-host OpenVMS Cluster SCA (system communications architecture) communications.
-

Multiple-Host SCSI Access

Multiple host access allows you to build highly available servers using shared access to SCSI storage. You can build a three-node OpenVMS Cluster system using a shared SCSI bus as the storage interconnect. You can also include the shared SCSI buses within a larger OpenVMS Cluster configuration. Using a quorum disk on the SCSI bus improves the availability of two- or three-node configurations.

The SCSI Standard

Rules governing the interactions between initiators and SCSI targets are defined by the SCSI standard. This standard also allows the host to communicate with the SCSI devices without having to manage the device-specific characteristics.



Note

- When connecting devices to a SCSI interconnect, each device on the interconnect must have a unique device ID. You may need to change a device's default device ID to make it unique. Refer to the device's owner's guide for details on changing the device ID.
- Default performance is 8-bit asynchronous (narrow mode).
- Data transfer is acknowledged by the device receiving the data.
- 16-bit and 32-bit wide options allow a doubling or quadrupling of the data rate. 32-bit is seldom implemented. 16-bit is referred to as wide.
- In synchronous mode, multiple data transfers can occur in succession followed by an acknowledgment from the device receiving the data.

All communications on a SCSI interconnect occur between two devices at a time; each pair of devices must negotiate to determine which of the optional features they will use. Most devices implement one or more of these options.

Devices

You can support up to eight devices on a SCSI bus. These include:

- Host adapters
- Peripheral controllers
- Discrete peripheral devices (disks, tape drives)

You assign device IDs (unique numbers between 0 and 7) by entering console commands or by setting jumpers or switches.

You can increase the number of devices by using a second level of device addressing using logical unit numbers (LUN). Each device ID can use up to eight LUNs (0-7) to address a single SCSI device as a multiple unit. The maximum number of LUNs per device ID is eight.

Performance

Narrow mode is the default mode of operation for all SCSI devices. It transfers 8 bits of data from one device to another. To enhance performance:

- Increase the amount of data transferred in parallel on the interconnect.
- Use synchronous data transfer.

The SCSI standard defines these modes for synchronous data transfer:

- Slow mode
Interconnect achieves up to 5 million transfers per second (also known as standard mode).
- Fast mode
Interconnect achieves up to 10 million transfers per second.

Distance

The signaling method used in the configuration determines the maximum length of the SCSI interconnect.

There are two types of electrical signaling used in SCSI interconnects:

- Single-ended signaling
 - Most common
 - Least expensive
 - Operates in either standard or fast mode
 - ◆ Standard transfers allow interconnects up to 6 meters in length.
 - ◆ Fast transfers allow interconnects up to 3 meters in length.
- Differential signaling
 - Higher signal integrity
 - SCSI bus can span distances up to 25 meters
 - Allows both standard and fast transfers regardless of the length of the SCSI bus

A DWZZA converter is a single-ended to differential converter. You can use this to connect single-ended and differential SCSI interconnect segments.

Differential segments:

- Overcome distance limitations of the single-ended interconnect
- Allow communication between single-ended and differential devices

**Note**

- In some cases the data transfer rate determines the maximum length of the SCSI interconnect.
 - When considering cable distances, be sure to include both internal and external cabling in your calculations.
 - DWZZA is strictly a signal converter. You don't need to assign a SCSI device ID to it. You can configure a maximum of two DWZZA converters in the path between any two hosts.
-

Cabling and Termination

Each single-ended and differential SCSI interconnect must have two terminators, one at each end. Specified maximum interconnect lengths are measured from terminator to terminator. Interconnect terminators are powered from the SCSI interconnect line call TERMPWR.

Devices attach to the interconnect by short cables (or etch) called stubs. Stubs must be short to maintain the signal integrity of the interconnect. The maximum stub lengths allowed are determined by the type of signaling used by the interconnect.

Single-ended interconnects allow a maximum stub length of .1 meters. Differential allow a maximum of .2 meters. The minimum distance between stubs on a single-ended interconnect is .3 meters.

When using the host system's internal SCSI adapter, you must configure the system at the end of the single-ended SCSI segment because the internal SCSI cable length exceeds the allowable SCSI stub length.

**Note**

- Each Digital host adapter and enclosure supplies the TERMPWR interconnect line, so that as long as one host or enclosure is powered on, the interconnect remains terminated.
 - Terminate single-ended and differential buses individually, even when you use a DWZZA converter.
 - When you extend the SCSI bus beyond an existing terminator, it is necessary to disable or remove that terminator.
 - Host systems do not have to be configured at the ends of the bus segment when an add-on SCSI adapter is used.
-

SCSI OpenVMS Cluster Hardware Configurations

You can connect up to three hosts on a shared SCSI interconnect. Each host in turn can be connected to one, two, or three shared SCSI interconnects.

Nonshared, configurable SCSI buses are limited only by the number of slots that are available on the host bus.

You can build systems using:

- Add-on SCSI adapters
- Internal SCSI adapters



Note

The Adaptec 1740/1742 SCSI controlled (PB2HA-SA) is not supported on AlphaServer 2100 systems having more than one gigabyte of memory. If the controller is connected to such a system, the following message appears on the operator's console:

```
%PKJDRVR-E- The direct DMA window does not map all of  
memory. Port is going OFF LINE
```

Add-On SCSI Adapters

Using KZPAA Adapters

You can use KZPAA adapters in shared SCSI bus configurations. The internal adapter is used to connect devices that cannot be shared, such as SCSI tape, floppies, or CD-ROM drives.



Note

- For KZPAA adapters, storage is configured using BA350, BA353, and HSZ40 StorageWorks enclosures.
 - Using StorageWorks enclosures, you can shut down individual systems without losing access to the disks.
-

Building a Basic System

A basic configuration typically allows the use of tapes, floppies, CD-ROMs, and disks with nonshared files on internal buses.

If you need to remove a system and still keep your OpenVMS Cluster system operational, you can build your system using DWZAA converters. If you need continuous access to data if a SCSI interconnect fails, do both of these:

- Add a redundant SCSI interconnect.
- Shadow the data.

Removing a Server

You can build a system that allows a server to be removed using a DWZAA converter (a SCSI bus isolator). You would want to remove a system for repair or for maintenance. Being able to do this gives you a higher level of availability.

DWZAA converters provide additional SCSI bus length capabilities. Using these converters, you can use a single-ended device to connect to a bus that uses differential signaling. You can use converters with the internal SCSI adapter or the optional KZPAA adapters.

DWZAA converters are available as:

- Standalone components
- Desktop components
- StorageWorks compatible building blocks



Note

SCSI bus configurations that use differential signaling can span distances up to 25 meters. A single-ended configuration only spans 3 meters if using fast mode data transfer.

HSZ40 Controllers

You can build a system that allows additional features and performance by using an HSZ40 controller. The HSZ40 is a high-performance differential SCSI controller that can be connected to a differential SCSI bus. It supports up to 42 SCSI devices.

You can configure the HSZ40 on a shared SCSI bus that includes DWZZA single-ended to differential converters.

Disk devices configured on the HSZ40 controllers can be combined into RAID sets to further enhance performance and provide high availability.

Additional Enclosures

Using a configuration where DWZZA converters are placed between systems with single-ended signaling and a differential-cabled SCSI bus allows you to add more enclosures or have greater separation.

You can connect up to three hosts on a multiple-host SCSI bus in OpenVMS Version 7.0.

Internal SCSI Adapters

Benefits

You can build a multiple-host SCSI OpenVMS Cluster configuration with two systems using internal adapters that are joined by a single SCSI cable.

This configuration provides the benefits of multiple-host SCSI systems that use external adapters, as well as being an inexpensive configuration.

You can expand this type of configuration to provide:

- Improved performance
- Availability
- Scaling



Note

An example of an external adapter is a fully shared disk with twice the serving performance of a single system.

Limitations

Only using internal SCSI clusters in a multiple-host system has these limitations:

- Internal cables exceeding 3 meters disallows the uses of fast mode data transfer.
- Removing either of the AlphaServer systems while the remaining cluster stays active leaves the bus nonterminated.
- Quorum is lost when one of the enclosures goes down if these are the only members of the cluster.
- Tape or CD--ROM drives cannot be used.

You can eliminate some of these limitations by using DWZZA controllers, additional SCSI adapters, and additional storage enclosures.

SCSI Environment Parameters

Parameter	Description
<code>bootdef_dev</code> <i>device_name</i>	Specifies the default boot device to the system.
<code>boot_osflags</code> , <code>root_number</code> , <code>bootflag</code>	Contains information that is used by the operating system to determine optional aspects of a system bootstrap (for example, conversational bootstrap).
<code>pk*0_disconnect</code>	Allows the target to disconnect from the SCSI bus while the target acts on a command. When this parameter is set to 1, the target is allowed to disconnect from the SCSI bus while processing a command. When the parameter is set to 0, the target retains control of the SCSI bus while acting on a command.
<code>pk*0_fast</code>	Enables SCSI adapters to perform in fast SCSI mode. When this parameter is set to 1, the default speed is set to fast mode; when the parameter is 0, the default speed is standard mode.
<code>pk*0_host_id</code>	Sets the SCSI device ID of host adapters to a value between 0 and 7.
<code>scsi_poll</code>	Enables console polling on all SCSI interconnects when the system is halted.
<code>control_scsi_term</code>	Enables and disables the terminator on the integral SCSI interconnect at the system bulkhead (for some systems).

Review of System Directories

Overview

The OpenVMS operating system stores its image files, data files, and other files in system directories that can be referenced using systemwide logical names.

Because all OpenVMS systems use these names, it is important for the system manager to be familiar with them.

Topics

In this section the following topics are discussed:

- OpenVMS system directories

OpenVMS System Directories

System Directories

Each system directory has its own logical name, specifying:

- The system device
- The specific system directory

To use the directory name to refer to a file in a system directory:

- Use the logical name SYS\$SYSROOT as the device name.
- SYS\$SYSROOT is defined as a **concealed root directory**.

Concealed root directories allow flexible handling of multiple versions of OpenVMS software on the same system.

Concealed Root Directories

The system root directories have the following format, where *n* is a hexadecimal number from 0 through FFFF.

- [SYS*n*]

A typical nonclustered OpenVMS system will only have one system root directory, [SYS0].

Ensure that you do *not* mix the use of root directory names and top-level directory names.

Incorrect:	SYS\$SYSROOT:[SYS0.SYSEX]LINK.EXE
Correct:	SYS\$SYSROOT:[SYSEX]LINK.EXE



Note

Additional notes on rooted directories:

- Roots [SYS0] through [SYSF] can be specified in R5.
- Roots [SYS10] through [SYSFFFF] can only be used for satellites.
- The boot node informs the satellite which root to boot.
- Use the following NCP command to display information about the rooted directory being used by a satellite node:

```
NCP>SHOW CHARACTERISTICS NODE TINY
```

```
Node Volatile Characteristics as of 13-NOV-1999 13:51:49
```

```
Remote node = 61.450 (TINY)
```

```
Hardware address = 08-00-2B-0F-48-3C
```

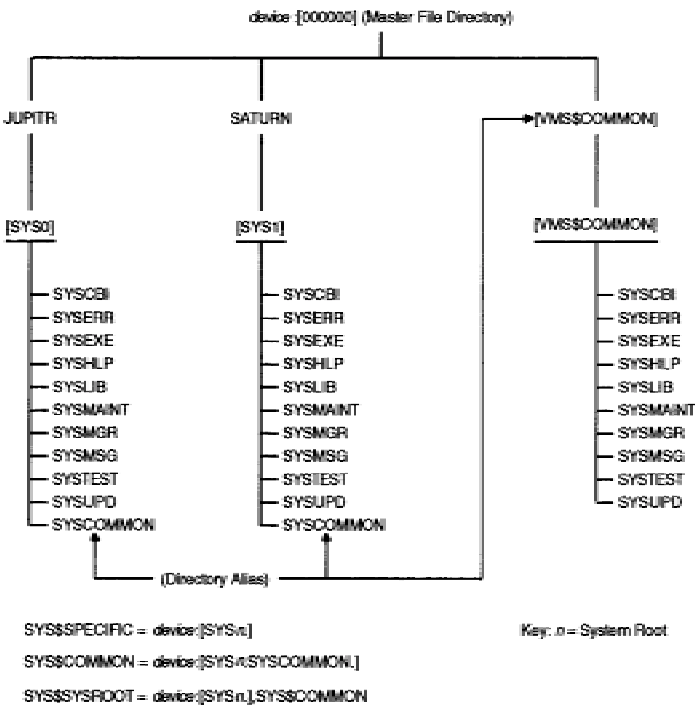
```
Tertiary loader = SYS$SYSTEM:TERTIARY_VMB.EXE
```

```
Load Assist Agent = SYS$SHARE:NISCS_LAA.EXE
```

```
Load Assist Parameter = DISK$VAXVMSRL052:<SYS17.>
```

Figure: System Directory Tree

The following figure shows a system directory tree:



OpenVMS System Directories and Their Contents

The following table describes the system directories located under the concealed root directory SYS\$SYSROOT.

This directory...	Has this logical name...	And contains:
[SYSEXE]	SYS\$SYSTEM	Executable images, user authorization file, site-independent startup and shutdown command procedures, system parameter files, print form definitions, paging file, swap file, crash dump file
[SYSLIB]	SYS\$LIBRARY SYS\$SHARE	Macro libraries, object libraries, shareable image libraries, shareable images
[SYSHLP] [SYSHLP.EXAMPLES]	SYS\$HELP SYS\$EXAMPLES	Help libraries Sample source programs such as device drivers and system services
[SYSHLP.VMSDOC]	SYS\$VMSDOC_TEXT	Printable files for certain OpenVMS documentation manuals
[SYSMSG] [SYSMGR]	SYS\$MESSAGE SYS\$MANAGER	Error message files Site-specific startup and shutdown command procedures, operator's log, accounting file
[SYSMAINT]	SYS\$MAINTENANCE	Hardware diagnostics (not distributed with an OpenVMS system)
[SYSTEST]	SYS\$TEST	UETP command procedures, images, data
[SYSERR] [SYSUPD]	SYS\$ERRORLOG SYS\$UPDATE	Error log file Command procedures used for system installation, upgrade, and update
[SYSCBI]	SYS\$INSTRUCTION	Computer-based instruction software
[SYS\$LDR]	SYS\$LOADABLE_IMAGES	Loadable executive images and device drivers
[SYS\$STARTUP]	SYS\$STARTUP	Startup command procedures

Other System Logical Names

The following table describes other commonly used system logical names.

This logical name...	Refers to:
SYS\$SYSDEVICE	The system device. <pre>\$ SHOW LOG SYS\$SYSDEVICE "SYS\$SYSDEVICE" = "\$1\$DUA0:" (LNM\$SYSTEM_TABLE)</pre>
SYS\$SYSROOT	The rooted logical name for your node. Points to both node-specific and shared files on the system disk. <pre>\$ SHOW LOGICAL SYS\$SYSROOT "SYS\$SYSROOT" = "\$1\$DUA0:[SYS2.]" (LNM\$SYSTEM_TABLE) = "SYS\$COMMON:"</pre>
SYS\$SPECIFIC	The node-specific portion of the system disk. <pre>\$ SHOW LOGICAL SYS\$SPECIFIC "SYS\$SPECIFIC" = "\$1\$DUA0:[SYS2.]" (LNM\$SYSTEM_TABLE)</pre>
SYS\$COMMON	That portion of the system disk shared by all nodes <pre>\$ SHOW LOGICAL SYS\$COMMON "SYS\$COMMON" = "\$1\$DUA0:[SYS2.SYSCOMMON.]" (LNM\$SYSTEM_TABLE)</pre>

Obtaining Information About Volumes

Overview

Some characteristics of disks and tapes can be changed by the system manager. To manage disk and tape volumes, the system manager needs to understand the characteristics and the commands used to change them.

Topics

In this section the following topics are discussed:

- Volume types and characteristics
- Displaying device characteristics

Volume Types and Characteristics

Public Volumes

Public volumes are volumes (disks or tapes) to which all users on the system have access. File and directory protection features restrict access to files and directories located on the public volume. Public volumes may contain operating system files or user files, or both.

Public volumes are not owned by any user or group exclusively. Instead, the system account owns them.

The system manager is responsible for maintaining public volumes.

Private Volumes

A volume owned by a single user or group of users is called a **private volume**. A private volume has an owner user identification code (UIC).

On occasion, a user may need to perform work on a device to which no one else has access. By creating a private volume and mounting it on a device assigned exclusively to the user process, the user can work without interference from other users.

A user can move files from one system to another without the use of a network by copying the files to a private tape volume.

Private tape volumes can also be used to preserve files that need to be protected from unauthorized access.

Volume Label

A volume label is an alphanumeric name encoded on a volume.

- Tape volume labels can contain a maximum of six characters.
- Disk volume labels can contain a maximum of 12 characters.

The label remains in effect until it is explicitly changed. Removing the volume from the drive does not affect the label.

The system checks the volume label before allowing certain operations to occur. If the volume label in the command does not match the volume label on the disk or tape, the user is notified and must override the volume label to continue. Volume ownership or special privileges may be required to override a volume label.

Other Information Encoded on a Volume

Each volume has an owner specified in standard UIC format encoded on the volume. Public volumes have the UIC [1,1].

The user name is a string of 1 through 12 alphanumeric characters recorded on the volume. The default name is the user name of the process that created the volume.

Physical Volume Labels

Removable volumes (disks and tapes) should have a paper label affixed to the outside of the volume casing. The physical label should not be confused with the labels encoded on the volume, although in most cases, they contain the same information. The physical label allows the volume to be identified without using the computer.

Removable volumes to be stored in large media libraries are usually given a physical reel number that is written on the physical label.

Displaying Device Characteristics

Use the SHOW DEVICE command to display information about devices on a system.

SHOW DEVICE Command

To display this information...	Use this command:
All devices on the system	SHOW DEVICE
List of only disk devices	SHOW DEVICE D
Complete status of a device (useful for determining the configuration of disks in a cluster)	SHOW DEVICE/FULL
Files that are currently open on this node	SHOW DEVICE/FILES

Example: SHOW DEVICE Command

The following example displays information about all devices having a device code that begins with D.

```
$ SHOW DEVICE D
```

Device Name	Device Status	Error Count	Volume Label	Free Blocks	Trans Count	Mnt Cnt
DOGS\$DKA500:	Mounted	0	WINDOWS	0	1	13
DSA0:	Mounted	0	OPENVMS_60	174093	374	13
\$1\$DUA0: (BARNUM)	ShadowSetMember	0	(member of DSA0:)			
\$1\$DUA1: (BARNUM)	Mounted	0	USER1	132489	3	13
\$1\$DUA2: (BARNUM)	Mounted	0	USER2	281922	16	13
\$1\$DUA3: (BARNUM)	OnLine	0				

Example: Locally Connected Disks

In the following example, assume that \$1\$DUA2 is local to node BARNUM and that the command is entered on BARNUM.

```
$ SHOW DEVICE/FULL $1$DUA2
```

Disk \$1\$DUA2: (BARNUM), device type RA82, ❶ is online, mounted, ❷ file-oriented device, shareable, served to cluster via MSCP Server, error logging is enabled.

Error count	00❷	Operations completed	128302
Owner process	""	Owner UIC	[SYSTEM]❸
Owner process ID	00000000	Dev Prot	S:RWPL,O:RWPL,G:R,W❹
Reference count	18	Default buffer size	512
Total blocks	1216665❹	Sectors per track	57
Total cylinders	1423	Tracks per cylinder	15
Allocation class	1		
Volume label	"USER2"	Relative volume number	0
Cluster size	3	Transaction count	16
Free blocks	281922❺	Maximum files allowed	152083
Extend quantity	5	Mount count	13
Mount status	System	Cache name	"_DSA0:XQPCACHE"
Extent cache size	64	Maximum blocks in extent cache	28192
File ID cache size	64	Blocks currently in extent cache	8118
Quota cache size	0	Maximum buffers in FCP cache	285
Volume owner UIC	[SYSTEM]	Vol Prot	S:RWCD,O:RWCD,G:RWCD,W:RWCD

Volume status: subject to mount verification, write-through caching enabled.
Volume is also mounted on RNLNG, FLYING, BAILEY, LIONS, TIGERS, BEARS, OHMY, COTTON, CANDY, CLOWNS, FUN, DOGS❻

Example Notes: Locally Connected Disks

This example answers questions, including the following:

- ❶ What type of disk is it?
- ❷ Is it mounted on the local system?
- ❸ Is it mounted on any other system in the cluster?
- ❹ How big is it (in 512-byte blocks)?
- ❺ How many blocks are free?
- ❻ What is the owner UIC and volume protection?
- ❼ Has it generated any hardware errors since the system was started up?



Note

The mount count tells you the total number of systems that have mounted the disk. See also the list of nodes, other than BARNUM, that have mounted the disk.

Note too that the **Allocation class** does not display for devices beginning with a zero. For devices that begin with a zero, the applicable host name is displayed.

Example: Remote Disks

In the following example, assume that \$1\$DUA2 is local to node BARNUM and we enter the SHOW DEVICE command from BAILEY.

```
$ SHOW DEVICE/FULL $1$DUA2
```

Disk \$1\$DUA2: (BARNUM), device type RA82, is online, mounted, file-oriented device, shareable, available to cluster, error logging is enabled.

Error count	0	Operations completed	126
Owner process	""	Owner UIC	[SYSTEM]
Owner process ID	00000000	Dev Prot	S:RWPL,O:RWPL,G:R,W
Reference count	1	Default buffer size	512
Total blocks	1216665	Sectors per track	57
Total cylinders	1423	Tracks per cylinder	15
Host name	"BARNUM" ^❶	Host type, avail	VAX 4000-300, yes ^❷
Allocation class	1		
Volume label	"USER2"	Relative volume number	0
Cluster size	3	Transaction count	1
Free blocks	297615	Maximum files allowed	152083
Extend quantity	5	Mount count	13
Mount status	System	Cache name	"_DSA0:XQPCACHE"
Extent cache size	64	Maximum blocks in extent cache	29761
File ID cache size	64	Blocks currently in extent cache	435
Quota cache size	0	Maximum buffers in FCP cache	212
Volume owner UIC	[SYSTEM]	Vol Prot	S:RWCD,O:RWCD,G:RWCD,W:RWCD

Volume status: subject to mount verification, write-through caching enabled.
Volume is also mounted on BARNUM, RNLNG, FLYING, LIONS, TIGERS, BEARS,
OHMY, COTTON, CANDY, CLOWNS, FUN, DOGS

Example Notes: Remote Disks

Additional information you might want to know about a remote disk:

- ❶ Which cluster node is the disk connected to?
- ❷ What type of node is it connected to?

Example: ISO 9660 CD-ROM Disks

In the following example we enter the SHOW DEVICE command to a CD--ROM disk that is in ISO 9660 format.

Disk DOGS\$DKA500:, device type RRD42, is online, mounted, file-oriented device, shareable, available to cluster, error logging is enabled.

Error count	0	Operations completed	26
Owner process	""	Owner UIC	[0,0]
Owner process ID	00000000	Dev Prot	S:RWPL,0:RWPL,G:R,W
Reference count	1	Default buffer size	512
Total blocks	795432	Sectors per track	4
Total cylinders	33143	Tracks per cylinder	6
Host name	" DOGS"	Host type, avail VAXstation	4000-60,yes
Volume label	"WINDOWS"	Relative volume number	1
Cluster size	0	Transaction count	1
Free blocks	0	Maximum files allowed	0
Extend quantity	0	Mount count	13
Mount status	System	ACP process name	"DKA500CACP"②

Volume status: ISO 9660.❶

Volume is also mounted on RNGLNG, FLYING, BAILEY, LIONS, TIGERS, BEARS, OHMY, COTTON, CANDY, CLOWNS, FUN, DOGS

Members of this volume set are DOGS\$DKA500: (rvn 1).

Example Notes: ISO 9660 CD-ROM Disks

This SHOW DEVICE output is similar to that on OpenVMS disks except that the display provides the ISO 9660 format indication and XQP data is not displayed.

❶ Indicates the ISO 9660 format

② The OpenVMS process that handles the non OpenVMS ISO 9660 disk format

**Note**

This example shows the correct ACP that runs when the ISO 9660 CD--ROM disk is run.

```

OpenVMS 7.2 on node BARNUM 21-NOV-1999 15:19:03.64    Uptime 5 17:21:53
  Pid  Process Name  State Pri    I/O      CPU      Page flts  Pages
2020081 SWAPPER      HIB   16      0  0 00:00:09.58      0      0
20200405 MARSH        CUR    4   5472  0 00:00:23.08     17811    456
20200086 CONFIGURE    HIB   10     46  0 00:00:00.08      127    253
20200088 IPCACP       HIB   10     46  0 00:00:00.09      115    231
20200089 ERRFMT       HIB    8   4745  0 00:00:18.68      115    230
.
.
.
202004B4 NFS$SERVER    LEF   10    427  0 00:00:02.64      2079   1663 N
20200836 DKA500CACP❶     HIB   10     20  0 00:00:00.27       706    488
20200AB8 ELOG_COLLECT_1 HIB    8      0  0 00:00:00.02        64    157
20200A39 NOTES$0000D_0* HIB    6   6606  0 00:00:15.39     29171   147 N
2020043E DECW$TE_043E  LEF    5  13341  0 00:01:02.81     4492   5039
2020043F Da BEARS      LEF    4  10944  0 00:00:05.51      922    535
.
.
.

```

- ❶ This is the OpenVMS process that handles the ISO9660 format. If this were a High Sierra format disk, the process name would be DKA500DACP.

Open Files on a Volume

You may need to determine if any open files exist on a volume:

- You cannot dismount a disk volume because it has open files on it.
- The output shows you which files are open and by which process.
- A user cannot open a shared data file because another user already has it open.
- Look for that file in the output and note which user's process has it open.

Listing Open Files

Use the SHOW DEVICE command to show which files on a particular device are open by processes on the local system.

Format:

SHOW DEVICE/FILES device-name

Example: Listing Open Files

The following example lists the open files on the device DUA0:.

```
$ SHOW DEVICE/FILES DUA0:
```

```
Files accessed on device DSA0: on 22-NOV-1999 09:56:21.40
```

Process name	PID	File name
	00000000	[000000]INDEXF.SYS;1
	00000000	[SYSCOMMON.SYSEXEX]F11BXQP.EXE;1
	00000000	[SYS0.SYSEXEX]PAGEFILE.SYS;1
	00000000	[SYSCOMMON.SYS\$LDR]SYS.EXE;3
	00000000	[SYSCOMMON.SYS\$LDR]EXEC_INIT.EXE;1
.		
.		
.		
	00000000	[SYSCOMMON.SYSEXEX]FAL.EXE;1
	00000000	[SYSCOMMON.SYSEXEX]NETSERVER.EXE;1
	00000000	[SYSCOMMON.SYSLIB]NMLSHR.EXE;1
CONFIGURE	20200086	[SYSCOMMON.SYSEXEX]CONFIGURE.EXE;1
IPCACP	20200088	[SYSCOMMON.SYSEXEX]IPCACP.EXE;1
ERRFMT	20200089	[SYSCOMMON.SYSEXEX]ERRFMT.EXE;1
CACHE_SERVER	2020008A	[SYSCOMMON.SYSEXEX]FILESERV.EXE;1
CLUSTER_SERVER	2020008B	[SYSCOMMON.SYSEXEX]CSP.EXE;1
OPCOM	2020008C	[SYSCOMMON.SYSEXEX]OPCOM.EXE;1
CLUSTER_SERVER	2020008B	[SYS0.SYSEXEX]SYS\$INCARNATION.DAT;1
AUDIT_SERVER	2020008D	
		[SYSCOMMON.SYSEXEX]AUDIT_SERVER.EXE;1
JOB_CONTROL	2020008E	
		[SYSCOMMON.SYSEXEX]JBC\$JOB_CONTROL.EXE;1
JOB_CONTROL	2020008E	[SYS0.SYSMGR]ACCOUNTNG.DAT;1
JOB_CONTROL	2020008E	
		[SYSCOMMON.SYSEXEX]QMAN\$MASTER.DAT;16
QUEUE_MANAGER	2020008F	
		[SYSCOMMON.SYSEXEX]QMAN\$MASTER.DAT;16
.		
.		
.		

Preparing Volumes for Use

Overview

System managers are responsible for creating and maintaining public volumes on the system. Individual users may take responsibility for the private volumes they own; however, the system manager is responsible for assigning private volumes and making them initially accessible to the user.

Only devices that conform to Digital Storage Architecture are considered in this section.

Topics

In this section the following topics are discussed:

- Allocating a device
- Initializing a volume
- Mounting a volume
- Removing volumes
- Creating a public volume
- Creating a volume set

Allocating a Device

Some operations are best performed when your process has exclusive access to a device. An example of this is volume initialization. Your process should be the only process to have access to the device that holds the volume to be initialized. This will prevent other processes from interfering with the initialization.

Gaining Exclusive Access to a Device

The ALLOCATE command provides your process with exclusive access to a device (disk drive or tape drive) until you deallocate the device by:

- Entering the DEALLOCATE command
- Entering a command that automatically deallocates the device
- Terminating your process

Format:

ALLOCATE device-name[:] [logical-name[:]]

The ALLOCATE command can optionally associate a logical name with the device. This logical name can then be used to refer to the device until the device is deallocated. The use of such logical names reduces typing and prevents errors.

Allocating a Specific Device

To allocate a specific device, specify the physical device or a logical name that translates to the name of a physical device.

Example:

```
$ ALLOCATE _DMB2
%DCL-I-ALLOC, _DMB2: allocated
```

The system response indicates that the device was successfully allocated.

Allocating a Generic Device

The `ALLOCATE` command allows you to specify a generic device name to request any available device of a particular type.

For example, a device name of `DU` would cause the system to allocate any available disk drive with the disk type of `DU`.

```
$ ALLOCATE DU DISK_DRIVE
```

The logical name `DISK_DRIVE` is equated to the device that is allocated.

If more than one device is specified, the first available device is allocated.

```
$ ALLOCATE MT,MF: TAPE_DRIVE
```

The `ALLOCATE` command in this example requests the allocation of any tape device whose name begins with `MT` or `MF`, and assigns the logical name of `TAPE_DRIVE` to the device.

Initializing a Volume

Before files or data can be written to a disk or tape, the volume must be initialized. Initializing a volume does the following:

- Invalidates all existing data on the volume, if any, and creates a new file structure
- Writes a label on the volume to identify it
- Defines the owner UIC and the protection for the volume

Before entering the INITIALIZE command, you should allocate the device that you will use to initialize the volume.

In the case of fixed (nonremovable) disks, a device is dedicated to one disk. With removable media such as tapes and removable disks, the media must be physically loaded onto the allocated device before it can be initialized.

Requirements for Initializing a Volume

To initialize a volume, one of the following must be true:

- The volume has never been used.
- The owner UIC of the volume matches your UIC.
- You have VOLPRO privilege.

INITIALIZE Command

Use the INITIALIZE command to format a volume and write a label on the volume.

Format:

INITIALIZE *device-name*[:*volume-label*]

- The *device-name* specifies the name of the device on which the volume to be initialized is physically mounted. A logical name can be used in place of the device-name.
- The *volume-label* specifies the identification to be encoded on the volume. The volume-label for a disk can contain up to 12 alphanumeric characters or 6 alphanumeric characters for a tape volume.



Warning

Initialization destroys all existing data on a disk. Do not enter this command on a used disk without first making a backup copy of all the data on the disk, unless you are willing to take the chance that the data will never be needed again.

Initializing a Public Disk

To initialize a public disk, use the qualifier /SYSTEM. This sets the owner UIC to [1,1] and protection defaults to complete access by all ownership categories (S:RWCD,O:RWCD,G:RWCD,W:RWCD).

Example:

```
$ INITIALIZE/SYSTEM DUA2: USER8
```

Initialization of a public disk requires a system UIC or SYSPRV privilege.

Initializing a Private Disk

The following command initializes the disk DUA2: as a private disk.

```
$ INITIALIZE DUA2: TESTDISK
```

Because no owner was specified, the UIC of the process entering the command is assigned as the owner. All categories of access by all categories of users will be permitted.

Initializing a Group Disk

Initialize a volume with the /GROUP qualifier to apply a protection code of RWCD to all UIC categories and to give the disk an owner UIC of [group,0].

```
$ INITIALIZE/GROUP DBA1:
```

Use the qualifiers /GROUP/NOSHARE to give the disk a protection code of RWCD to all UIC categories except WORLD, which will have no access.

```
$ INITIALIZE/GROUP/NOSHARE DBA1:
```

Changing Protection on Initialization

Use the /PROTECTION qualifier to apply a different protection code at the time a disk is initialized.

```
$ INITIALIZE/PROTECTION=(S:RWCD,O:RWCD,G:R,W) DBA1:
```

System and owner UIC will be given READ and WRITE access, regardless of what is specified with the qualifier. This example initializes a private disk giving group members READ access. World UICs have no access.

Initializing a Tape Volume

Tape volumes must be initialized before they can be used. The format used for tapes is specified by standards for informational interchange. These standards facilitate the portability of data written to tape. Tapes written on one brand of computer equipment can be read on the equipment of another vendor.

The following example initializes a tape and assigns the volume label SOURCE.

```
$ INITIALIZE MTB1: SOURCE
```

Mounting a Volume

After an initialized volume (tape or disk) is physically loaded on a device (disk drive or tape drive), use the Mount utility to make the disk or tape volume available to the OpenVMS operating system.

The Mount utility does the following:

- Ensures that the device has not been allocated to another user.
- Ensures that a volume is physically loaded on the device specified.
- Ensures that the label on the volume matches the label specified.
- If the volume is a magnetic tape, MOUNT also checks the volume accessibility field of the volume label.

Disks must be mounted each time the system is booted. Commands to mount public disks are usually included in the site-specific SYSTARTUP_VMS.COM.

MOUNT Command

Use the MOUNT command to invoke the Mount utility.

Format:

MOUNT *device-name*[:] [*label*] [*logical-name*[:]]

- The *device-name* specifies the physical device name or logical name of the device on which the volume is physically loaded.
- The *label* is the name of the volume specified when the volume was initialized. The system compares the volume label in the command with the volume label on the volume.

Mounting a Public Disk

Use the /SYSTEM qualifier to mount a public disk.

The following command mounts a public disk with the volume label USER8.

```
$ MOUNT/SYSTEM DUA1: USER8
```

Mounting a Disk Clusterwide

Disks can also be mounted so that they can be accessed from all nodes in an OpenVMS Cluster system. To mount a disk clusterwide, use the /CLUSTER qualifier. Only system and group volumes can be mounted clusterwide. If you specify the /CLUSTER qualifier without either the /SYSTEM or /GROUP qualifier, the default is /SYSTEM. You must also use a cluster device-naming convention. Use either **node\$device-name** or **allocation-class\$device-name** as required by your configuration. The following example mounts the disk DMA1 on HSC node DOPEY clusterwide.

```
$ MOUNT/CLUSTER DOPEY$DMA1: SNOWWHITE
```

Setting Protection on MOUNT

When you mount a disk, you can change the established protection codes if you own the volume or have the VOLPRO privilege. Do this with the /PROTECTION qualifier.

```
$ MOUNT/PROTECTION =(S:RWCD,O:RWCD,G:R,W:R) DBA1: WORKDISK
```

Removing Volumes

When you are finished using a volume:

- Dismount the volume to break the connection between your process and the contents of the volume.
- Deallocate the device to free it for use by other processes.

Dismounting a Volume

Use the DISMOUNT command to:

- Close a mounted disk or tape volume for further processing.
- Deassign the logical name associated with the volume.

Format:

DISMOUNT device-name[:]

If you want to keep the volume on line when you dismount it, use the /NOUNLOAD qualifier.

```
$ DISMOUNT/NOUNLOAD MUA0:
```

All open files on the volume must be closed before the actual dismount can be done. If the volume was mounted to be accessible by other users, the volume remains mounted until all processes using that volume dismount it or terminate.

The DISMOUNT command does not deallocate a device that was allocated with the ALLOCATE command.

Deallocating a Device

Use the DEALLOCATE command to free a device for use by other users. Deallocating a device does not delete a logical name assigned by the ALLOCATE command.

Format:

DEALLOCATE device-name[:]

Example:

```
$ DEALLOCATE DMA1:
```

**Note**

Previously, the DCL command DISMOUNT performed a set of simple tests before attempting to dismount a Files-11 volume. These tests determined that:

- The volume was really mounted.
- The volume was not the system disk.
- The user had the necessary privileges to dismount the volume.

If these tests ran successfully, the volume was marked for dismount and the command returned a success status. In reality the actual dismount did not take place until the volume was actually idle. The volume could stay in the marked-for-dismount state for some time until all files were closed.

DISMOUNT checks for conditions that prevent the command from completing. These include:

- Installed swapping files and paging files
- Installed images
- Devices spooled to the volume
- Open user files

If any of the above conditions exist, the DISMOUNT command does **not** mark the volume for dismount, but instead, returns a message that explains why the volume cannot be dismounted.

Creating a Public Volume

Steps for Creating a Public Volume

To create a new public volume, follow these steps:

- Select an unused disk.
- Allocate the device to your process.
- Initialize the disk using the /SYSTEM qualifier.
- Mount the disk using the /SYSTEM qualifier.

Initializing a Public Volume

To initialize a public volume, initialize the disk using the /SYSTEM qualifier.

Format:

INITIALIZE/SYSTEM device-name[:] label

This will:

- Write the specified *label* on the volume
- Set the owner UIC to [1,1]
- Set the volume protection to S:RWCD, O:RWCD, G:RWCD, W:RWCD

This command requires a system UIC or SYSPRV privilege.

Mounting a Public Volume

Use the MOUNT/SYSTEM command to mount a public volume.

Format:

MOUNT/SYSTEM device-name[:] label [logical-name[:]]

This will:

- Place the logical name for the device in the system logical name table
- Allow all users to access the volume

This command requires SYSNAM privilege.

Example: Preparing a Public Disk

The following example shows how to prepare a public disk:

```
❶ $ SHOW DEVICE $1$DUA2

Device   Device Error Volume
Name     Status Count Label
$1$DUA2: (TIDY) Online 0

❷ $ ALLOCATE $1$DUA2 PUBDISK
%DCL-I-ALLOC, _$1$DUA2: allocated

❸ $ INITIALIZE/SYSTEM PUBDISK MYVOL
❹ $ MOUNT/SYSTEM PUBDISK MYVOL
%MOUNT-I-MOUNTED, MYVOL mounted on _$1$DUA2: (TIDY)
$
$ SHOW LOGICAL/SYSTEM DISK*

(LNM$SYSTEM_TABLE)

"DISK$MYVOL" = "$1$DUA2:"
" DISK$TESTSPACE" = "$1$DUA1:"
"DISK$VAXVMSRL052" = "$1$DUA0:"
```

Example Notes: Preparing a Public Disk

Notes on the Preparing a Public Disk example include:

- ❶ Display the status of a particular device.
- ❷ Allocate the device to your process. The system displays a message informing you that the device is allocated. The system assigns the logical name PUBDISK to the device and records the assignment in your process logical name table.
- ❸ Initialize the disk with the label MYVOL. The /SYSTEM qualifier sets the owner UIC to [1,1] and sets the protection code to (S:RWCD,O:RWCD,G:RWCD,W:RWCD). These values remain in effect until the volume is initialized again.
- ❹ Mounting the disk with the /SYSTEM qualifier deallocates the device and allows all users access to the volume without mounting it.

Creating a Volume Set

Volume Sets

If the files or user directories become too large to fit on any one volume, you can create a **volume set**. This consists of two or more disk volumes bound together with the MOUNT/BIND command.

The OpenVMS operating system treats a volume set as one large volume by:

- Storing files on any volume in the set that has free space
- Attempting to use space evenly over all volumes in a set

Procedure for Creating a Volume Set

Use the following procedure to create a disk volume set:

- Allocate the necessary devices, and physically load the volumes on the devices.
- Initialize each new volume in the set.
- Use the MOUNT/BIND command to create the volume set.

Example:

```
$ MOUNT/BIND=MASTER_SET DUA1: , DUA2: PAYVOL1, PAYVOL2
```

The name of the volume set in the example is MASTER_SET. PAYVOL1 (DUA1:) becomes the root volume because it is listed first. Its master file directory (MFD) contains the directory structure for the entire volume set. The second volume, PAYVOL2 (DUA2:), is bound to the first volume to create the volume set.

Dismounting a Volume Set

To dismount an entire volume set, use the DISMOUNT command and specify any one of the devices containing a member of the volume set. If it is a public volume set, this command requires SYSNAM privilege.

Format:

DISMOUNT device-name[:]

To dismount a single volume of a volume set, include the /UNIT qualifier.

DISMOUNT/UNIT device-name[:]

When you dismount a volume set, the system deletes the logical names created during the mount procedure.

Example: Creating a Volume Set

The following example shows how to create a public volume set (USER_SET) starting with an existing volume (USER1) as the root volume.

```
❶ $ ALLOCATE DRA2 DEV1
   %DCL-I-ALLOC, _DRA2: allocated
   $ ALLOCATE DRA3 DEV2
   %DCL-I-ALLOC, _DRA3: allocated
❷ $ INITIALIZE/SYSTEM DEV2 USER2
❸ $ MOUNT/SYSTEM/BIND=USER_SET DEV1,DEV2 USER1,USER2
   %MOUNT-I-MOUNTED, USER1 mounted on _DRA2:
   %MOUNT-I-MOUNTED, USER2 mounted on _DRA3:
❹ $ SHOW LOGICAL/SYSTEM D*
```

```
(LNM$SYSTEM_TABLE)
```

```
"DBG$INPUT" = "SYS$INPUT:"
"DBG$OUTPUT" = "SYS$OUTPUT:"
"DDP$DIS" = "SYS$MANAGER:DDP.DIS"
"DISK$USER1" = "DRA2:"
"DISK$USER2" = "DRA3:"
"DISK$USER_SET" = "DRA2:"
"DISK$VAXVMSRL052" = "DUA0:"
"DTR$LIBRARY" = "SYS$SYSROOT:[DTR]"
```

```
$
```

```
❺ $ COPY WORK1:[BROWN]EXAMP5.COM
   _To: DISK$USER_SET:[SMITH]EXAMP5.COM
$ DIRECTORY DISK$USER_SET:[SMITH]
```

```
Directory DISK$USER_SET:[SMITH]
```

```
EXAMP5.COM;1
```

```
Total of 1 file.
```

```
❻ $ DISMOUNT DEV1
$ SHOW LOGICAL/SYSTEM D*
```

```
(LNM$SYSTEM_TABLE)
```

```
"DBG$INPUT" = "SYS$INPUT:"
"DBG$OUTPUT" = "SYS$OUTPUT:"
"DDP$DIS" = "SYS$MANAGER:DDP.DIS"
"DISK$VAXVMSRL052" = "DUA0:"
"DTR$LIBRARY" = "SYS$SYSROOT:[DTR]"
```

```
$
```

Example Notes: Creating a Volume Set

Notes on the Creating a Volume Set example include:

- ❶ Allocate two disk devices for the volumes USER1 and USER2, and give them the logical names DEV1 and DEV2 respectively. After allocating the devices, the user loads the volumes (USER1 and USER2) into their respective drives. (USER1 is an existing volume; USER2 is a new volume.)
- ❷ Initialize USER2 to delete old files and to create a Files-11 structure. The /SYSTEM qualifier sets the owner UIC to [1,1] and the protection code to (S:RWCD,O:RWCD,G:RWCD,W:RWCD).
- ❸ Bind the volumes into the disk volume set, USER_SET. The /SYSTEM qualifier makes the volume set public. The root volume (USER1) contains the directory structure for the entire volume set.
- ❹ A logical name was not included in the MOUNT command. Therefore, the system creates the logical names DISK\$USER1, DISK\$USER2, and DISK\$USER_SET by default.

If you mount USER_SET at a later time, you could include a logical name for the volume set in the MOUNT command. In that case, the system would not create the default logical name in the form **DISK\$volume_set_name**. For example, the following command includes the logical name USERS, which can be used as the name of the volume set in subsequent commands.

```
$ MOUNT/SYSTEM DEV1, DEV2 USER1, USER2 USERS
```

- ❺ Copy a file from another disk to a directory on the volume set. (The [SMITH] directory already existed on the USER1 volume. It is now a directory on the volume set.) The system stores the file on the volume in the set that has the most unused space.
- ❻ Dismount the entire volume set by dismounting one member of the volume set. Because the volume set was mounted using the /SYSTEM qualifier, the devices have already been deallocated. Therefore, they are now available for other users.
- ❼ When you dismount a volume set, the system deletes the logical names it created during the mount procedure.

Handling Common Volume Problems

Overview

Common volume problems include shortage of disk space, incorrect file ownership, fragmentation of disk space, incorrectly dismounted disks, and incorrect volume labels.

Topics

In this section the following topics are discussed:

- Setting file characteristics
- Defragmenting a disk
- Rebuilding incorrectly dismounted disks
- Specifying a new volume label

Setting File Characteristics

One way to manage the use of disk volumes is to modify certain characteristics of files. The system manager can specify expiration dates for files, set version limits to enable automatic purging of files, and change file ownership if it has been set incorrectly.

SET FILE Command

Use the SET FILE command to modify the characteristics of files.

Format:

- The *file-spec* specifies one or more files to be modified.
- If you specify more than one file, separate them with commas.
(Wildcard characters are allowed.)

File Characteristics That Can Be Modified

The following table lists some of the file characteristics you can modify using the SET FILE command.

To modify this characteristic...	Use this qualifier:
Expiration date	/EXPIRATION_DATE /NOEXPIRATION_DATE Assigns an expiration date to the specified files. Later, you can use DELETE/EXPIRED to delete files whose expiration date has passed.
Version limit	/VERSION_LIMIT[=n] Specifies the maximum number of versions allowed for the specified file. If the version limit is exceeded, the earliest version of the file is deleted from the directory without notification to the user.
Owner	/OWNER_UIC[=uic] Specifies an owner UIC for the file. The default is the UIC of the current process. The OpenVMS operating system records the use of disk space on a UIC basis. Therefore, changing the owner UIC of a file changes which disk quota entry the space is recorded under.



Note

To change the ownership of the NEILSEN directory enter the SET FILE /OWNER=NEILSEN \$1\$DIA200:[000000]NEILSEN.DIR.

Modifying the Action of the SET FILE Command

The following table lists some of the qualifiers you can use to modify the action of the SET FILE command.

To request this action...	Use this qualifier:
Select files belonging to a particular UIC.	/BY_OWNER[=uic] Selects only those files whose owner UIC matches the specified UIC. The default is the UIC of the current process.
Exclude a specified file (or files) from the SET FILE operation.	/EXCLUDE=(file-spec[,...]) Wildcard characters are allowed.
Confirm the SET FILE operation.	/CONFIRM /NOCONFIRM Controls whether a request is issued before each SET FILE operation to confirm that the operation should be performed on that file. The default is /NOCONFIRM.
Log the operation.	/LOG Displays the file specification of each file modified as the SET FILE command executes.

Example: Changing File Characteristics

The following example changes the ownership, expiration date, and version limit of a file.

❶ \$ DIRECTORY/FULL TEST.COM

Directory WHYSO\$DUA0:[USER1]

TEST.COM;1 File ID: (1620,1,0)

❷ Size: 1/3 Owner: [USER1]

Created: 9-NOV-1999 14:25:03.66

Revised: 22-NOV-1999 16:00:20.22 (3)

❸ Expires: <None specified>

Backup: <No backup recorded>

File organization: Sequential

File attributes: Allocation: 3, Extend: 0, Global buffer count: 0

❹ No version limit

Record format: Variable length, maximum 12 bytes

Record attributes: Carriage return carriage control

RMS attributes: None

Journaling enabled: None

File protection: System:RWED, Owner:RWED, Group:RE, World:

Access Cntrl List: None

Total of 1 file, 1/3 blocks.

\$

❺ \$ SET FILE /OWNER=USER2 /EXPIRATION_DATE=0 /VERSION_LIMIT=3-
\$_ /LOG TEST.COM

❻ %SET-I-MODIFIED, WHYSO\$DUA0:[USER1]TEST.COM;1 modified

\$

❼ \$ DIRECTORY/FULL TEST.COM

Directory WHYSO\$DUA0:[USER1]

TEST.COM;1 File ID: (1620,1,0)

❽ Size: 1/3 Owner: [USER2]

Created: 9-NOV-1999 14:25:03.66

Revised: 22-NOV-1999 11:21:37.92 (4)

❾ Expires: 22-NOV 1999 00:00:00.00

Backup: <No backup recorded>

File organization: Sequential

File attributes: Allocation: 3, Extend: 0, Global buffer count: 0

❿ Version limit: 3

Record format: Variable length, maximum 12 bytes

Record attributes: Carriage return carriage control

RMS attributes: None

Journaling enabled: None

File protection: System:RWED, Owner:RWED, Group:RE, World:

Access Cntrl List: None

Total of 1 file, 1/3 blocks.

\$

Example Notes: Changing File Characteristics

Notes on the Changing File Characteristics example include:

- ❶ Display full directory information about TEST.COM.
- ❷ The owner UIC of the file is [USER1].
- ❸ The file has no expiration date.
- ❹ The file has no version limit.
- ❺ Change several characteristics of the file and request that the name of the file be displayed when the operation takes place.
- ❻ The log message.
- ❼ Redisplay the directory information.
- ❽ The owner UIC of the file is now [USER2].
- ❾ The file now has an expiration date that defaults to the current date because 0 was specified as the date.
- ❿ The file now has a version limit of 3.



Note

Instructor Note

To limit all the files in a new directory to three versions, use the SET DIRECTORY command.

```
$ SET DIRECTORY/VERSION_LIMIT=3 [USER3]
```

The SET DIRECTORY command affects only the files created after it is entered. To affect currently existing files in a directory, use the SET FILE command.

Note that you can override the effects of the SET DIRECTORY command for any given file in a directory by using the SET FILE command. For example, if the version limit for a directory is three versions, you can use the SET FILE command to increase the limit for FILE.DAT to five versions.

Defragmenting a Disk

Disk Fragmentation

Disk files are not written contiguously on the disk. When a file is written to the disk, the OpenVMS operating system looks for space to write the file. The file may be written to two or three areas of the disk. Over time the available space on the disk becomes more and more fragmented and, as a result, files written to the disk are also fragmented.

Using BACKUP to Defragment a Disk

You can use the BACKUP command to defragment a disk. The contents of a badly fragmented disk should be copied to another disk of the same or larger size using the /IMAGE qualifier. This will make all of the files contiguous and leave a large contiguous chunk of free space.

- The output volume must be mounted using the /FOREIGN qualifier.
- No files on the disk should be open by any other users.

Handling Files Marked NOBACKUP

When performing an image backup to disk, all files on the disk must be copied. Some users may have flagged a file NOBACKUP because there is no need to back up a file that is available from another source. This is done using the SET FILE/NOBACKUP command.

```
$ SET FILE/NOBACKUP file_name.txt
```

When defragmenting a disk, you must include these flagged files in the copy operation. Use the BACKUP qualifier /IGNORE=NOBACKUP to include these files.

Example: Using BACKUP to Defragment a Disk

In the following example, the fragmented disk DMA2: is backed up to DMA1:.

```
$ MOUNT/FOREIGN DMA1:
%MOUNT-I-MOUNTED, mounted on NODE$DMA1:
$ BACKUP/IMAGE/IGNORE=NOBACKUP DMA2: DMA1:
.
.
.
```


Rebuilding Incorrectly Dismounted Disks

Incorrectly Dismounted Disks

A disk will be incorrectly dismounted if there is a system failure, or if someone powers off a disk drive without dismounting the disk. If a disk is incorrectly dismounted, free space and storage allocation inconsistencies may result.

Rebuild the disk to correct problems resulting from incorrect dismounting.

Automatically Rebuilding a Disk

Rebuilding takes place automatically when you mount an improperly dismounted disk.

- Use the MOUNT/NOREBUILD command to prevent rebuilding.
- The MOUNT/NOREBUILD command should be used in the startup command procedure to mount each disk in an OpenVMS Cluster system. If this command is not used, processes using the disk will hang until the rebuild completes.
- If the system disk is rebuilt at startup time, and the system disk is also being served by the MSCP server, the cluster can hang indefinitely.

To prevent this situation on any system that serves a system disk or that boots from a served system disk, use AUTOGEN to set the SYSGEN parameter ACP_REBLDSYSD to 0. This parameter defaults to 0 for satellites; you must set it for boot servers.

Explicitly Rebuilding a Disk

Use the SET VOLUME command to explicitly rebuild a disk. If you are not the owner of the volume, this command requires a system UIC or SYSPRV privilege.

Format:

SET VOLUME/REBUILD device-name[:]

If the disk does not need to be rebuilt, this command has no effect.

Sample commands:

```
$ SET VOLUME/REBUILD SYS$SYSDEVICE
$ SET VOLUME/REBUILD $1$DUA1:
```

Delaying a Disk Rebuild Operation

There is no risk to data integrity by not rebuilding a disk at startup time. The worst consequence is that some free space is not available until the disk is rebuilt.

Disks can be rebuilt at a more convenient time, such as in a batch job at an off-hour.

Specifying a New Volume Label

One response to an "incorrect volume label" error is to specify a new volume label. For example, when using standalone backup to make a full backup of a disk, the volume label of the target disk is changed to match that of the source disk if no new label is specified.

Using the SET VOLUME Command

To specify a new volume label, use the SET VOLUME command.

Format:

SET VOLUME device[:] /LABEL=label

Privileges required for the command are:

- Write access to the index file of the volume
- SYSPRV if not using the SYSTEM account

Example: Specifying a New Volume Label

In the following example, the volume label is changed from WOOD to PAPER.

```
$SHOW DEVICE $1$DUC7
```

Device Name	Device Status	Error Count	Volume Label	Free Blocks	Trans Count	Mnt Cnt
\$1\$DUC7:	(BARNUM) Mounted	0	WOOD	1140414	1	12

```
$SET VOLUME $1$DUC7 /LABEL=PAPER  
$SHOW DEVICE $1$DUC7
```

Device Name	Device Status	Error Count	Volume Label	Free Blocks	Trans Count	Mnt Cnt
\$1\$DUC7:	(BARNUM) Mounted	0	PAPER	1140414	1	12

```
$
```

Summary

Concepts

Review of OpenVMS Device Concepts

- Every device has a unique name in the format of **ddcu**.
- Any OpenVMS node in a cluster can mount an MSCP served device and access it as if it were a local device.
- Two formats are used in naming devices in an OpenVMS Cluster system:
- **node\$device**
- **\$allocation-class\$device**

SCSI Clusters

- One of the benefits of OpenVMS Cluster systems is that multiple computers can simultaneously access storage devices connected to an OpenVMS Cluster storage interconnect. Together, these systems provide high performance and highly available access to storage.
- With OpenVMS Alpha Version 6.2 or later, you can connect multiple Alpha hosts to a single SCSI bus to share direct access to SCSI storage devices.

Review of System Directories

- OpenVMS operating system files are stored in special disk locations. These files and locations are named using standard systemwide logical names.

Obtaining Information About Volumes

- Public volumes are accessible to all system users.
- Private volumes are accessible by a single user or group of users.
- Each volume has a volume label, owner, and user name encoded on the volume.
- Use the SHOW DEVICE command to obtain information about volumes.

Preparing Volumes for Use

- Use the ALLOCATE command to gain exclusive access to a device.
- Before you can write files or data to a disk or tape, the volume must be initialized.
- A volume must be mounted before its contents can be accessed.
- Dismount a volume to make its contents unavailable for further access.
- Deallocate a device to free it for use by other users.
- To create a public volume, initialize and mount it with the /SYSTEM qualifier.
- A volume set consists of two or more volumes logically bound together. The OpenVMS operating system treats a volume set as one large volume.

Handling Common Volume Problems

- Use the SET FILE command to modify file characteristics including expiration date, version limit, and owner UIC.
- Use the BACKUP command to defragment a disk.
- If a disk is incorrectly dismounted, free space and storage allocation inconsistencies may result. To correct this, the disk should be rebuilt.
- Use the SET VOLUME command to specify a new volume label.

Commands

Obtaining Information About Volumes

SHOW DEVICE

Displays information about devices.

Preparing Volumes for Use

ALLOCATE

Obtains exclusive access of a device for a process.

INITIALIZE

Formats a disk or tape volume and writes a label on the volume.

MOUNT

Makes a disk or tape volume available for processing.

DISMOUNT

Closes a mounted disk or tape volume for further processing and deassigns the logical name associated with the device.

DEALLOCATE

Makes an allocated device available to other processes, but does not deassign any logical name associated with the device.

INITIALIZE/SYSTEM device-name[:] label

Initializes a public volume.

MOUNT/BIND

Creates a volume set.

DISMOUNT device-name[:]

Dismounts a volume set.

DISMOUNT/UNIT device-name[:]

Dismounts one volume of a volume set.

Handling Common Volume Problems

SET FILE

Modifies the characteristics of one or more files.

BACKUP

Used to defragment a disk by making files contiguous.

MOUNT

Automatically rebuilds an improperly dismounted disk.

SET VOLUME

Explicitly rebuilds an improperly dismounted disk, or specifies a new volume label.

Introduction to System Customization

Module 5

About This Module

Introduction

The OpenVMS operating system offers a variety of ways that the system manager can customize the system. The system environment can be customized by modifying system startup files and system parameters. The system manager can customize the environment for all users or for specific groups of users by using login command procedures.

This module introduces these customizable components and describes the steps to take to modify them.

Objectives

To perform basic system customization tasks, you should be able to:

- Name the standard system startup files and describe their functions.
- Describe the functions of the various login command procedures used on an OpenVMS system.
- Display and manipulate system parameters.
- Update the DECnet remote node database.

Customizing System Startup Files

Overview

Several command procedures are invoked when an OpenVMS system boots. To customize the computing environment, you need to know the order in which these files execute and the functions they perform.

System startup files should be modified when:

- The system is first installed
- New physical resources are added
- Software products are installed
- There is a change in system resources

Topics

In this section the following topics are discussed:

- Optional files
- Site-independent startup file
- Configuring devices
- Defining systemwide logical names
- Installing page and swap files
- Performing site-specific startup functions

Optional Files

Adding or Deleting Files

The OpenVMS operating system lets you customize the size of the operating system by deleting or adding optional system files, including support for DECwindows software. This is particularly valuable for small systems or systems with limited disk space.

Depending on the system you are using, you can add or delete files in one of the following ways:

- On VAX systems, by using the OpenVMS tailoring utilities:
 - VMSTAILOR: applies to optional system files
 - DECW\$TAILOR: applies to support for DECwindows

To delete from or add files to the system disk, you identify classes and subclasses of operating system files. You can delete or add an entire class or selected subclasses of files within a class. If you delete files, you can add them again at any time using VMSTAILOR or DECW\$TAILOR and your operating system distribution media.
- On Alpha systems, by using the POLYCENTER Software Installation utility (PCSI). You add or delete files by selecting or deselecting PCSI options and suboptions.

To invoke PCSI for this purpose, use either of the following methods:

- Enter the command `$ PRODUCT RECONFIGURE`
- Invoke the PCSI Motif interface and select the Reconfigure option on the Mode menu.

Site-Independent Startup File

STARTUP.COM

The site-independent startup file SYS\$SYSTEM:STARTUP.COM contains commands needed to initialize any OpenVMS system. To understand the operations performed in system startup, you should know what STARTUP.COM does.

This command procedure should *not* be modified. If a command in STARTUP.COM does not create the environment that you want, you should override that command with a command in the site-specific startup files rather than change the command in STARTUP.COM.

A database file named VMS\$PHASES.DAT determines the order of the phases of the startup procedure. It is a sequential list of the phases that STARTUP.COM starts; it includes a series of four basic phases (INITIAL, CONFIGURE, DEVICE, and BASEENVIRON) needed to start the operating system, followed by a series of phases for layered products. Do *not* modify VMS\$PHASES.DAT. To start up correctly, the system requires that the contents of this file remain intact.

Sequence of Operations for Startup

STARTUP.COM controls the following sequence of operations:

1. Definition of logical names needed for basic operations, and installation of images listed in SYS\$MANAGER:VMSIMAGES.DAT.
2. Executes SYS\$MANAGER:SYCONFIG.COM.
3. Automatically configures the device driver database, locates all standard devices attached to the system, and loads and connects their device drivers with one of the following commands:

- On VAX systems, the SYSGEN command AUTOCONFIGURE ALL
- On Alpha systems, the SYSMAN command IO AUTOCONFIGURE

If the symbol STARTUP\$AUTOCONFIGURE_ALL is defined as 0 or FALSE in SYS\$MANAGER:SYCONFIG.COM, this step is not performed.

4. SWAPFILE1.SYS, if present, is installed.
5. The CONFIGURE process starts.
6. SYS\$MANAGER:SYLOGICALS.COM defines systemwide logical names.
7. SYS\$MANAGER:SATELLITE_PAGE.COM mounts the satellite's local disk and installs the page and swap files on the satellite's local disk.
8. SYS\$MANAGER:SYPAGEFILES.COM installs page and swap files.
9. SYS\$MANAGER:SYSECURITY.COM mounts or defines any disks that hold security auditing log files or local security archive files.
10. SYS\$MANAGER:SYSTARTUP_VMS.COM executes site-specific customization commands.

**Note**

SWAPFILE1.SYS is reserved for use on a tailored system. If you have created a file called SWAPFILE1.SYS you must add a SYSGEN INSTALL command to SYPAGSWPFILES.COM to install it.

CLUSTER_CONFIG.COM creates the command procedure SATELLITE_PAGE.COM in the satellite's [SYSx.SYSEXE] directory on the boot server's system disk. SATELLITE_PAGE.COM performs the following functions:

- Mounts the satellite's local disk
 - Installs the page and swap files on the satellite's local disk
-

Component Files Used by STARTUP.COM

SYSS\$SYSTEM:STARTUP.COM uses a series of component files that accomplish many functions. These functions include:

- Assigning logical names required by certain OpenVMS system software components
- Assigning logical names to the OpenVMS system directories
- Starting up system processes such as:
 - JOB_CONTROL
 - QUEUE_MANAGER
 - OPCOM
 - ERRFMT
- Installing known images
- Connecting all standard devices
- Calling the site-specific startup command procedures

These component files are located in a set of directories associated with the systemwide logical name SYSS\$STARTUP.

How STARTUP.COM Starts the System

STARTUP.COM starts up the system using four basic phases:

- INITIAL
- CONFIGURE
- DEVICE
- BASEENVIRON

STARTUP.COM uses three data files, located in SYSS\$STARTUP:

- VMS\$PHASES.DAT determines the order of the phases during the startup procedure.
- VMS\$VMS.DAT is the data file for starting the base operating system environment.
- VMS\$LAYERED.DAT is the data file for layered products that are installed at system startup.

Do Not Modify Site-Independent Files

All major releases and some minor releases of the OpenVMS operating system delete and replace `STARTUP.COM`. The system manager should *not* modify this file.

Never modify `VMS$PHASES.DAT` or `VMS$VMS.DAT`. These files contain important site-independent information that can change from one operating system release to the next.

Configuring Devices

SYCONFIG.COM

SY\$MANAGER:SYCONFIG.COM is a site-specific command procedure invoked by STARTUP.COM. This file is used to connect special devices to the system and to load their I/O drivers.

Use of SYCONFIG.COM

You can add commands to SY\$MANAGER:SYCONFIG.COM to connect and configure nonstandard devices or unusual device settings each time the system starts:

- On VAX systems, SYSGEN CONNECT commands
- On Alpha systems, SYSMAN IO CONNECT commands

You can also optionally place MOUNT commands in this file. However, most sites mount the remaining disks using SYSTARTUP_VMS.COM.

When SYCONFIG.COM completes, control is returned to STARTUP.COM. STARTUP.COM automatically connects all remaining devices and loads their I/O drivers.

Example: SYCONFIG.COM

On VAX systems, you can enter SYSGEN commands like the following to connect nonstandard devices:

```
$!  
$!SY$STARTUP:SYCONFIG.COM  
$!  
$! Set virtual terminals....  
$!  
$ RUN SY$SYSTEM:SYSGEN  
CONNECT VTA0/NOADAPTER/DRIVER=TTDRIVER  
CONNECT TDA0/NOADAPTER/DRIVER=TDDRIVER  
EXIT  
$ EXIT
```


On Alpha systems, you can enter SYSMAN commands like the following:

```
$!  
$!SYS$STARTUP:SYCONFIG.COM  
$!  
$! Set virtual terminals....  
$!  
$ RUN SYS$SYSTEM:SYSMAN  
IO CONNECT FTA0:/DRIVER=SYS$FTDRIVER/NOADAPTER  
EXIT  
$ EXIT
```

**Note**

After SYCONFIG.COM completes execution, STARTUP.COM executes the SYSGEN command AUTOCONFIGURE ALL on VAX systems; on Alpha systems, STARTUP.COM executes the SYSMAN command IO AUTOCONFIGURE. Either command automatically configures all known devices attached to the system according to the parameters set in the SYSGEN parameter file.

To suppress autoconfiguration, include the following command as the last line in SYCONFIG.COM:

```
$ STARTUP$AUTOCONFIGURE_ALL == 0
```

In configuring devices, STARTUP.COM creates a detached process that detects any devices attached to an HSC unit and loads their corresponding drivers. Note, however, that if you set STARTUP\$AUTOCONFIGURE_ALL to zero in SYCONFIG.COM, the CONFIGURE section of STARTUP.COM will not execute.

To ensure correct configuration of devices connected to an HSC unit, Compaq recommends adding the following lines to the end of SYCONFIG.COM:

```
$ STARTUP$AUTOCONFIGURE_ALL == 0  
$ @SYS$SYSTEM:STARTUP CONFIGURE  
$ EXIT
```

Defining Systemwide Logical Names

SYLOGICALS.COM

SYSS\$MANAGER:SYLOGICALS.COM is a site-specific command procedure invoked by STARTUP.COM. This file is used to define systemwide logical names. A template file is supplied by Compaq.

How to Create System Logical Names

Use the ASSIGN or DEFINE command to create system logical names.

Format:

ASSIGN/SYSTEM equivalence logical-name

DEFINE/SYSTEM logical-name equivalence

- These commands require SYSNAM privilege.
- Use the /NOLOG qualifier with either DEFINE or ASSIGN to reduce the amount of printout on the console terminal during system startup.

How to Delete System Logical Names

Use the DEASSIGN command to delete system logical names.

Format:

DEASSIGN/SYSTEM logical-name

This command requires SYSNAM privilege.

Standard Logical Names

The following table lists some standard logical names defined in SYLOGICALS.COM.

This logical name...	Represents:
SYS\$SYLOGIN	The systemwide login command procedure
SYS\$ANNOUNCE	A line of text, or the name of a file containing text, that is displayed when a user presses the Return key to log in
SYS\$WELCOME	A line of text, or the name of a file containing text, that is displayed after a user successfully logs in

Executive-Mode Logical Names

Security considerations require you to define some systemwide logical names as executive-mode logical names. Whenever you decide to define a systemwide logical name, consider its use by privileged images, or in situations when the integrity of the system could be compromised by incorrect logical names.

Logical names for some system components and files must be defined in executive mode. During critical situations, only executive-mode and kernel-mode logical names are used by the operating system; supervisor-mode and user-mode logical names are ignored.

Examples of components and files you would set up executive-mode logical names for:

- Public disks and directories
- SYSUAF.DAT
- RIGHTSLIST.DAT
- VMSMAIL_PROFILE.DATA
- NETPROXY.DAT



Note

By default, the DECwindows loginout code performs a security check to note the presence of other clients during the initial connection to the server. If this check detects other clients, the DECwindows loginout process exits.

If you define DECW\$LOGIN_MULTIPLE in executive-mode in the system logical name table, the security check is bypassed. The translation of DECW\$LOGIN_MULTIPLE is unimportant to this security check. Only the existence of the logical name, not its value, is checked.

How to Create Executive-Mode Logical Names

To define an executive-mode logical name, use the ASSIGN or DEFINE command.

Format:

ASSIGN/SYSTEM/EXECUTIVE equivalence logical-name
DEFINE/SYSTEM/EXECUTIVE logical-name equivalence

Examples:

```
$ ASSIGN/SYSTEM/EXECUTIVE SECURE:[ADMIN]CLASS.DAT CLASSES
$ DEFINE/SYSTEM/EXECUTIVE CLASSES SECURE:[ADMIN]CLASS.DAT
```



Note

One executive-mode logical name commonly defined within SYLOGICALS.COM is SYSUAF, a pointer to the site user authorization file (SYSUAF.DAT).

```
$ DEFINE/SYSTEM/EXECUTIVE/NOLOG SYSUAF SYS$SYSTEM:SYSUAF.DAT
```

Defining the SYSUAF logical name in this way allows you to access SYSUAF.DAT (using the Authorize utility) without having to set SYS\$SYSTEM as your default directory.

Example: Assigning Site-Specific Logical Names

The following command procedure is a sample portion of SYLOGICALS.COM, the command procedure used to assign site-specific system logical names.

```
$!
$! Assign site-specific logical names
$!
$ ASSIGN/SYSTEM/EXECUTIVE/NOLOG DISK$USER:[PUBLIC]          SYS$PUBLIC
$ ASSIGN/SYSTEM/EXECUTIVE/NOLOG DISK$USER:[TOOLS]           SYS$TOOLS
$ ASSIGN/SYSTEM/EXECUTIVE/NOLOG "This is the MENTOR system"  SYS$ANNOUNCE
$ ASSIGN/SYSTEM/EXECUTIVE/NOLOG "@SYS$MANAGER:WELCOME.TXT"  SYS$WELCOME
$!
```

Example: SYLOGICALS.COM

The following is an example of a SYLOGICALS.COM:

```
$ !SYS$STARTUP:SYLOGICALS.COM
$ !
.
.
.
$ !
$ ! Define any site-specific logical names below:
$ !
$ ! This includes site-specific cluster common file definitions (previously
$ ! defined in the site-specific file SYENVIRON.COM).
$ ! The user should include definitions that define the location of
$ ! SYSUAF, NETUAF, VMSMAIL, RIGHTSLIST, NETNODE_REMOTE, and LMF$LICENSE.
$ ! Include a MOUNT/SYSTEM command for the disk that these files reside on.
$ ! See SYS$EXAMPLES:CLU_MOUNT_DISK.COM for the recommended method of doing
this.
$ !
$ node = f$getsyi("nodename")
$ !
$ ! Define clusterwide systems operations logicals
$ !
$ ASSIGN/SYSTEM/EXEC WORK4:[COMMON_SYSEX] SYS$COM_FILES
$ ASSIGN/SYSTEM/EXEC SYS$COM_FILES:SYSUAF.DAT SYSUAF
$ ASSIGN/SYSTEM/EXEC SYS$COM_FILES:NETPROXY.DAT NETPROXY
$ ASSIGN/SYSTEM/EXEC SYS$COM_FILES:RIGHTSLIST.DAT RIGHTSLIST
$ ASSIGN/SYSTEM/EXEC SYS$COM_FILES:VMSMAIL_PROFILE.DATA VMSMAIL_PROFILE
$ ASSIGN/SYSTEM/EXEC SYS$COMMON:[SYSEX]NETNODE_REMOTE.DAT NETNODE_REMOTE
$ !
$ ASSIGN/SYSTEM/EXEC 7 MAIL$SYSTEM_FLAGS
$ ASSIGN/SYSTEM/EXEC SYS$COMMON:[SYSMGR]SYLOGIN SYS$SYLOGIN !remove .com for
MCR CLI
$ ASSIGN/SYSTEM/EXEC "Software Course Development" PSM$ANNOUNCE ! Header to
printouts
$ ASSIGN/SYSTEM/EXEC -
"
Node: 'node'

UNAUTHORIZED ACCESS IS PROHIBITED
" -
SYS$ANNOUNCE
$ ASSIGN/SYSTEM/EXEC"@SYS$COMMON:[SYSMGR]WELCOME.TXT" SYS$WELCOME
$ ASSIGN/SYSTEM 0 SHUTDOWN$MINIMUM_MINUTES
$! Location of COMMON system reports
$ ASSIGN/SYSTEM/EXEC/TRANS=(TERM,CONCEAL) $1$DUA4:[SYS$INFO.] SYS$INFO
$!
$! Define the location of the WORK4:[TEMPLATE] directory
$ DEFINE/EXEC/SYSTEM TEMPLATES WORK4:[TEMPLATES]
$ !
$ ! Define logical name for DECgraph
$ ASSIGN/SYSTEM SYS$COMMON:[DECGRAPH] GRAPH$LIBRARY
```

```
$ !
$ !Define logical names specific to Satellite nodes
$ ! IF (''node'' .EQS. "MOPPET") .OR. (''node'' .EQS. "BONKRS") THEN $GOTO
CI_NODE
$ !DEFINE /SYSTEM /EXEC DOC$BATCH ''node'_DOC$BATCH"
$ !
$CI_NODE:
$EXIT
```

Installing Page and Swap Files

SYSPAGSWPFILES.COM

SY\$MANAGER:SYSPAGSWPFILES.COM is a site-specific command procedure invoked by STARTUP.COM. This file is used to install page and swap files on disks other than the system disk.

Installing Page and Swap Files

Before invoking SYSPAGSWPFILES.COM, the system activates the following files if they exist in SY\$SYSTEM:

- PAGEFILE.SYS
- SWAPFILE.SYS
- SYSDUMP.DMP

STARTUP.COM then invokes SYSPAGSWPFILES.COM.

To activate additional page and swap files, insert whatever commands are needed for the installation. These may include:

- MOUNT, to mount disks containing additional page and swap files
- SYSGEN, to create and/or install additional page and swap files

**Note**

SYSPAGSWPFILES.COM is invoked after the page and swap files are activated, allowing the system manager to direct the system to activate other page and swap files on other disks.

You might consider creating additional page and swap files if you have a large system with a large amount of memory and/or processes, or if you want to distribute paging and swapping activity across multiple disks.

Example: Installing Page and Swap Files

The following is a sample SYPAGSWPFILES.COM file, which installs a page file and a swap file.

```
$ SET NOON
$ SET NOCONTROL_Y
$ !
$ ! SYS$MANAGER:SYPAGSWPFILES.COM
$ ! Sample system page and swap file command procedure
$ !
$ MOUNT/SYSTEM $1$DUA4: PGSWP
$ RUN SYS$SYSTEM:SYSGEN
INSTALL $1$DUA4:[SYSEXE]PAGEFILE.SYS/PAGEFILE
INSTALL $1$DUA4:[SYSEXE]SWAPFILE.SYS/SWAPFILE
EXIT
$ EXIT
```


Performing Site-Specific Startup Functions

SYSTARTUP_VMS.COM

SY\$MANAGER:SYSTARTUP_VMS.COM is the final site-specific command procedure invoked by STARTUP.COM. A template file is supplied by Compaq.

Functions of SYSTARTUP_VMS Command Procedure

SYSTARTUP_VMS.COM is used to accomplish functions not covered by the other site-specific command procedures, such as:

- Mounting public disks
- Setting device characteristics
- Starting queues and enabling autostart for queues
- Installing known images
- Installing resident images (Alpha only)
- Setting up the OpenVMS InfoServer Client software
- Running the System Dump Analyzer (SDA)
- Purging unwanted operator log files
- Submitting batch jobs that are run at system startup time
- Creating systemwide announcements
- Starting up the LAT protocol software
- Starting up the DECnet for OpenVMS network
- Starting the DIBOL Message manager
- Defining the number of interactive users

Using Separate Site-Specific Command Procedures

Use separate site-specific command procedures to perform major functions. This enables you to:

- Keep SYSTARTUP_VMS.COM small and manageable.
- Edit a small file to change the effects of any startup function.
- Reduce the chance of accidental editing.
- Allow clean, separate execution of functions after the system has been started.

This capability is sometimes needed when certain failures occur and the function needs to be reactivated. An example of this is when, for some reason, the queue manager process must be restarted. When this happens, you need to restart the print and batch queues. If you put all the initialization and startup functions for system queues in a separate file, you only need to execute that particular file to restart your entire queue system.

Example: SYSTARTUP_VMS.COM

The following is an example of a SYSTARTUP_VMS.com file:

```
$ SET NOON
$ SET NOCONTROL_Y
$ !
$ ! SYS$MANAGER:SYSTARTUP_VMS.COM
$ ! Sample site-specific system startup command procedure
$ !
$ ! Create logical name for supporting command procedures
$ DEFINE/NOLOG STARTUP_PROCS SYS$SYSROOT:[SYSMGR.STARTUP]
$ !
$ ! Mount site-specific volumes
$ @STARTUP_PROCS:MOUNTDSK.COM
$ !
$ ! Set device characteristics
$ @STARTUP_PROCS:DEVICES.COM
$ !
$ ! Define and start print queues
$ @STARTUP_PROCS:START_PRNT_QUEUE.COM
$ !
$ ! Define and start batch queues
$ @STARTUP_PROCS:START_BATCH_QUEUE.COM
$ !
$ ! Install known images
$ @STARTUP_PROCS:INSTALL.COM
$ !
$ ! Start DECnet software
$ @SYS$MANAGER:STARTNET.COM
$ !
$ ! Start LAT network
$ @SYS$MANAGER:LTLOAD.COM
$ !
$ ! Create reports about the last system failure
$ @STARTUP_PROCS:REPORT_FAILURE.COM
$ !
$ ! Purge old versions of system log files
$ PURGE/KEEP=3 SYS$MANAGER:*.LOG
$ !
$ ! Set the maximum number of interactive users
$ STARTUP$INTERACTIVE_LOGINS == 40
$ !
$ ! Announce availability of the system to all terminals
$ SUBMIT STARTUP_PROCS:START_ANNOUNCE.COM
$ !
$ ! End of SYS$MANAGER:SYSTARTUP_VMS.COM
$ EXIT
```



Note

The SYSMAN STARTUP command is an alternate method of starting layered products. This command enters product startup procedures in a startup database so that the system starts them automatically.

Many layered products have dependencies that are too complex for 5-phase processes available in the layered product startup database file (whose logical name is STARTUP\$STARTUP_LAYERED). If so, customers can write separate command procedures that can be executed using the SYSMAN STARTUP command.

Customizing Login Command Procedures

Overview

The OpenVMS operating system allows you to specify the user's environment in the form of login command procedures.

Topics

In this section the following topics are discussed:

- Introduction to login command procedures
- Uses for login command procedures

Introduction to Login Command Procedures

Login Command Procedures

Login command procedures are executed each time an interactive process or batch job is created. They can be:

- User specific
- Group specific
- Systemwide
- Other

Location of Login Command Procedures

The operating system expects to find login command procedures in:

- System logical name SYS\$SYLOGIN (systemwide procedure)
- LGICMD field in each user's UAF record (specific procedure assigned by the system manager)
- Default user login file LOGIN.COM

Order of Execution of Login Command Procedures

The login command procedure execution sequence is as follows:

- If SYS\$SYLOGIN is defined, execute the command procedure it designates.
- If LGICMD contains the name of a command procedure, execute it.
- If LGICMD is blank, execute SYS\$LOGIN:LOGIN.COM.



Note

The operating system executes only one login procedure after the system login procedure. This is either the login procedure specified in the LGICMD field or, if the LGICMD field is blank, the default procedure (SYS\$LOGIN:LOGIN).

If there is no other login file, the operating system simply displays a prompt after it executes the system manager's login procedure.

Determining Execution Mode

Login command procedures are executed at the beginning of batch and network jobs, as well as before interactive jobs. You can use the lexical function F\$MODE in a login command procedure to determine the type of job that is executing. The procedure can then branch to sequences of commands appropriate for that type of job.

Uses for Login Command Procedures

Common uses of login command procedures include:

- Personal login only
- System and personal login
- System and group login
- Captive login

Personal Login Command Procedures

In a system where users execute only a personal login command procedure, the system logical name SY\$\$SYLOGIN is undefined.

Users create the file LOGIN.COM in their own login default directories to customize their own environments. In the user authorization file, LGICMD does not have to be defined.

System and Personal Login Command Procedures

To cause both a system and personal login command procedure to be executed when users log in:

- Create a command procedure to customize a common user environment for all users on the system.
- Define the system logical name SY\$\$SYLOGIN to point to this command procedure.
- Users can create a file, LOGIN.COM, in their own login default directories to customize their own environments. LGICMD does not have to be defined in the UAF record.

System and Group Login Command Procedures

To cause both a system and a group login command procedure to be executed when users log in:

- Create a command procedure to customize a common user environment for all users on the system.
- Define the system logical name SY\$\$SYLOGIN to point to this command procedure.
- Group managers can create a command procedure to customize a common user environment for all users in their group.
- In the user authorization file, each user's LGICMD is defined as the file created for that user's group.

Captive Login Command Procedures

When users log in to an account that has a captive login command procedure, the login procedure runs an application for them. They interact only with that application, and never reach DCL command level.

To set up a captive account:

- Create a login procedure that customizes the user environment and prevents the user from changing it.
- Set the LGICMD field for the account to specify the captive login command procedure.
- Set the CAPTIVE flag in the UAF record for the captive account.

The SYS\$SYLOGIN definition is optional. This affects captive and noncaptive users.

Customizing System Parameters

Overview

The data structures and files that the OpenVMS system needs for its operation occupy physical memory and disk storage. System performance is best when the sizes of the system data structures and files match the system resources.

When the system is installed or upgraded, the system parameters and file sizes are automatically set by AUTOGEN.COM, a command procedure supplied by Compaq. Once your system has been operational, you may need to adjust some of these values.

When the physical resources of a system are changed, system parameter values and file sizes should also be recomputed.

Topics

In this section the following topics are discussed:

- System parameters
- Tools for maintaining system parameters
- Reconfiguring the system with AUTOGEN
- Maintaining parameters with SYSMAN

System Parameters

OpenVMS operating **system parameters** control such things as:

- Sizes of OpenVMS data structures in memory
- Number of installed image files
- Sizes of system files

Default Parameter Values

To ensure that system parameter values reflect current hardware resources, the system automatically determines and records these hardware resources when the OpenVMS system software is installed.

After compiling a current list of the hardware, the system computes the appropriate parameter values. These parameters become the default parameter values for the system; they are listed under the column named CURRENT in the output display of parameters and are stored in the following default parameter files:

- On VAX systems, SYS\$SYSTEM:VAXVMSSYS.PAR
- On Alpha systems, SYS\$SYSTEM:ALPHAVMSSYS.PAR



Note

On VAX systems, the operating system allocates and deallocates memory for processes in pages of 512 bytes. Some system parameter values are allocated in units of pages, while others are allocated in units of pagelets. A page on an Alpha system can be 8 kilobytes (KB) (8192 bytes), 16 KB, 32 KB, or 64 KB. A pagelet is a 512-byte unit of memory.

When reviewing parameter values, especially those parameters related to memory management, be sure to note the units required for each parameter.

Installed Images

The system creates a default list of images to be installed at system startup based on the size of physical memory and other factors. This list is stored in SYSS\$MANAGER:VMSIMAGES.DAT.

System Files

Certain system files must reflect the system they are on. These are also customized when you install the system software to reflect the physical memory size and disk capacity of your system.

The following table lists the system files and their functions.

This file...	Performs this function:
Page file SYS\$SYSTEM:PAGEFILE.SYS	Manages virtual memory
Swap file SYS\$SYSTEM:SWAPFILE.SYS	Manages physical memory use
Dump file SYS\$SYSTEM:SYSDUMP.DMP	Saves a copy of physical memory when the system fails

Tools for Maintaining System Parameters

The following tools can be used to maintain system parameters:

- `SYSS$SYSTEM:SYSGEN.EXE` — utility
- `SYSS$UPDATE:AUTOGEN.COM` — command procedure (recommended method)
- `SYSS$SYSTEM:SYSMAN.EXE` — utility
- `SYSBOOT` — conversational startup

System Generation Utility, SYSGEN

The system generation utility (SYSGEN) is a system management tool that performs privileged system configuration functions:

- Creates and modifies system parameters (not recommended)
- Loads device drivers
- Creates/installs additional page and swap files

AUTOGEN Command Procedure

`SYSS$UPDATE:AUTOGEN.COM`, the AUTOGEN command procedure, performs the following functions:

- Determines hardware resources
- Computes system parameters
- Creates a list of image files to install
- Calculates size of page, swap, and dump files

When you boot your system during installation, AUTOGEN generates system parameters that are suitable for your hardware configuration. However, if you have an unusual hardware configuration or special workload requirements, you may want to modify some system parameters and rerun AUTOGEN.



Note

Compaq recommends the use of the AUTOGEN command procedure when modifying system parameters and creating and adding page and swap files. Use of SYSGEN is not recommended for this purpose.

System Management Utility, SYSMAN

The system management utility (SYSMAN) centralizes the management of nodes and OpenVMS Cluster environments. Rather than logging in to individual nodes and repeating a set of management tasks, SYSMAN allows you to define your management environment to be a:

- Particular node
- Group of nodes
- OpenVMS Cluster environment

With sufficient privilege, you can manipulate system parameters on other nodes from your local node. SYSMAN executes these tasks on all nodes in the target environment.

Conversational Startup

OpenVMS processors that use a boot command procedure permit you to perform a conversational startup. During a conversational startup, an experienced system manager can perform the following tasks:

- Specify a minimum startup.
- Select an alternate file as the source of system parameter values.
- Set and show system parameters.
- Specify an alternate site-independent startup procedure.



Note

In a conversational boot of Alpha systems using OpenVMS Version 6.2, the system may not write a new copy of the ALPHAVMSSYS.PAR file if you change system parameters during the boot. To permanently change them, invoke SYSGEN after booting and enter the commands `USE ACTIVE` and `WRITE CURRENT`.

Reconfiguring the System with AUTOGEN

AUTOGEN.COM

The AUTOGEN.COM command procedure establishes initial values for all configuration-dependent system parameters so that they match your particular configuration.

Functions of AUTOGEN.COM

The AUTOGEN command procedure runs automatically at system installation, and performs the following functions:

- Determines system hardware resources
- Records appropriate system configuration (to be established at startup)
- Sets system parameter values using SYSGEN utility
- Creates list of images to install
- Creates system files using SYSGEN utility
- Optionally reboots the system to allow new parameters to take effect

When You Should Run AUTOGEN

You should run AUTOGEN:

- During a new installation or upgrade.
- When system physical resources change.
- When system workload changes significantly.
- When you add a layered (optional) software product. The product installation guide tells you what parameters you need to adjust.
- When you install a shared image.
- To specify a system parameter value or file size manually.

You can also use AUTOGEN when you want to specify a system parameter or file size. Because the value of one parameter can affect the values of other parameters, AUTOGEN recalculates values and sizes whenever you manually change a single parameter value or system file size.

When AUTOGEN Changes Take Effect

The changes made by AUTOGEN do not take effect until the next time the system is rebooted. For this reason, you can optionally have AUTOGEN reboot the system or manually reboot the system at a more convenient time.

Running AUTOGEN

To run AUTOGEN, use the following syntax:

@SYSS\$UPDATE:AUTOGEN [start-phase] [end-phase] [mode]

- The *start-phase* parameter specifies the initial AUTOGEN operation to be performed.
- The *end-phase* parameter specifies the final AUTOGEN operation to be performed.
- The *mode* parameter specifies the execution mode, either FEEDBACK or NOFEEDBACK, to determine whether AUTOGEN uses dynamic feedback data in its calculations. (This parameter is only used with SAVPARAMS or GETDATA as the start phase.)

Privileges Required to Run AUTOGEN

To make changes to the current parameter values, you need write access to the parameter file:

- On VAX systems, SYSS\$SYSTEM:VAXVMSSYS.PAR
- On Alpha systems, SYSS\$SYSTEM::ALPHAVMSSYS.PAR

AUTOGEN Phases

AUTOGEN executes all of its phases in sequence, from the specified start phase through the specified end phase. The following table lists the AUTOGEN phases, in the order of execution.

This phase...	Performs this function:
SAVPARAMS	Records feedback data
GETDATA	Collects data required for calculation
GENPARAMS	Calculates parameter values and file sizes, and generates a list of images to install
TESTFILES	Displays calculated file sizes
GENFILES	Generates new files
SETPARAMS	Saves calculated parameters
SHUTDOWN	Shuts down the system
REBOOT	Reboots to allow new files and parameters to take effect

Use the AUTOGEN HELP command to obtain information about AUTOGEN and its phases.

\$ @SYSS\$UPDATE:AUTOGEN HELP

**Note**

AUTOGEN uses two data files, SYSS\$SYSTEM:AGEN\$ADDHISTORY.TMP and AGEN\$ADDHISTORY.DAT.

- These files are used in conjunction with the feedback mechanism to avoid accumulating the values of ADD_SYMBOLS found in MODPARAMS.DAT and VMSPARAMS.DAT from one invocation of AUTOGEN to the next.
- These files should not be deleted or modified.

AUTOGEN generates the feedback report file SYSS\$SYSTEM:AGEN\$PARAMS.REPORT to help you decide if you want to use the system parameter values and system file sizes that it calculates. Information in the feedback report includes the following:

- All parameters and system files directly affected by the feedback
 - Current and new values
 - The feedback used in each parameter calculation
-

Steps for Changing System Parameters Without Changing File Sizes

To change system parameters without changing file sizes:

- Enter the following command to instruct AUTOGEN to gather information, perform calculations, and generate reports:
`$ @SYS$UPDATE:AUTOGEN SAVPARAMS GENPARAMS`
- Review these files:
PARAMS.DAT (information required for AUTOGEN calculations)
SETPARAMS.DAT (calculated parameters)
AGEN\$PARAMS.REPORT (report on feedback data)
- If you want to change any parameters, edit MODPARAMS.DAT and rerun AUTOGEN as in step 1.
- If you are satisfied with the contents of SETPARAMS.DAT, enter the following command:
`$ @SYS$UPDATE:AUTOGEN SETPARAMS REBOOT`

This command updates the system parameter values and reboots the system to install these values.



Note

It is not necessary to have AUTOGEN reboot the system. This can be done later at a more convenient time using `SYS$SYSTEM:SHUTDOWN.COM`.



Note

Beginning with OpenVMS Version 6.2 on Alpha systems and Version 7.0 on VAX systems, symbol assignments in MODPARAMS.DAT preceded with a dollar sign (\$) are not logged in AGEN\$PARAMS.REPORT. You should use this method for symbols you don't care about to reduce file size.

OpenVMS Version 7.0 on VAX systems uses a new calculation of NPAGEDYN and NPAGEVIR. You can set these parameters in MODPARAMS.DAT to override the calculated values.

OpenVMS Version 7.0 on VAX systems uses new calculations for PQL_DWSDEFAULT, PQL_MWSDEFAULT, VIRTUALPAGECNT, and WSMAX. See the *Release Notes* for a discussion of how to obtain the old behavior.

OpenVMS Version 7.0 on Alpha systems with 128 MB or more of memory, the system will specify a compressed system dump file. You can override this by setting DUMPSTYLE to 0 or 1 in MODPARAMS.DAT.

Steps for Changing System Parameters and File Sizes

To change system parameters and file sizes:

1. Enter the following command to cause AUTOGEN to display its calculations of system file sizes:
`$ @SYS$UPDATE:AUTOGEN SAVPARAMS TESTFILES`
2. Examine the output from AUTOGEN regarding the file size calculations.
3. If you want to change file sizes, edit MODPARAMS.DAT and rerun AUTOGEN as in step 1.
4. If you are satisfied with the calculations made by AUTOGEN, continue by entering the following command to reboot the system with the new file sizes:
`$ @SYS$UPDATE:AUTOGEN GENPARAMS REBOOT`



Note

A system reboot is required for the changes to take effect. It is not necessary for AUTOGEN to do the REBOOT phase. The system can be rebooted at a more convenient time using the normal shutdown procedure.

Example: MODPARAMS.DAT

The following example displays the MODPARAMS.DAT file for the node DITTO. Note that each system manager has initialed his or her changes to the file.

```
!  
! MODPARAMS.DAT for node DITTO  
!  
SCSSYSTEMID = 2197  
SCSNODE = "DITTO "  
PAGEFILE = 60000  
ADD_GBLPAGES = 425+507+157      ! CMS, BLISS32 and ADA (FJM 9/13/91)  
ADD_GBLSECTIONS = 4+5+2        ! CMS, BLISS32 and ADA (FJM 9/13/91)  
LOCKIDTBL = 2048                ! For RDB (AB 9/25/91)  
RESHASHTBL = 16384              ! Raised for CDD 40+ (DE 2/22/92)  
MIN_VIRTUALPAGECNT = 136100     ! For VTX (VHM 5/25/92)
```

Maintaining Parameters with SYSMAN

SYSMAN

The System Management utility (SYSMAN) enables you to manipulate system parameters on all nodes in the management environment you define.

**Note**

Compaq recommends you use AUTOGEN to modify system parameters. To view system parameters for a group of nodes or to change parameters quickly, however, you can use SYSMAN.

SYSMAN PARAMETERS Command

Use the SYSMAN PARAMETERS command to manipulate system parameters and parameter files.

Format:

PARAMETERS subcommand

OPER privilege is required to run SYSMAN.

SYSMAN PARAMETERS Subcommands

The following table summarizes the SYSMAN PARAMETERS subcommands.

To perform this function...	Use this subcommand:
Initialize the current work area with system parameter values.	USE
Display the values of system parameters in the work area; as well as the default, minimum, and maximum values of the parameters and their units of measure.	SHOW
Modify the value of a system parameter in the work area. (Does not modify parameter files, current system parameter file, or active system parameters.)	SET
Write the system parameter values to a parameter file, to the current system parameter file, or to the active system in memory. (Requires SYSLOCK privilege.)	WRITE

Displaying Parameters

To display system parameters, use the SHOW subcommand of the SYSMAN PARAMETERS command.

Format:

PARAMETERS SHOW parameter-specification

Specify any of the following:

- Name of a single parameter
- Name of a group of parameters
- A period (.)

This is interpreted as a request for the parameter specified in the last PARAMETERS SET or PARAMETERS SHOW command.

The parameter name can be abbreviated, but it must be unique or SYSMAN selects the first parameter that matches.

Example: Displaying System Parameters

The following example uses the SYSMAN PARAMETERS command to display a group of active system parameters.

```
❶ $ SET PROCESS/PRIVILEGE=OPER
$ RUN SYS$SYSTEM:SYSMAN
❷ SYSMAN> SET ENVIRONMENT/NODE=BARNUM
%SYSMAN-I-ENV, current command environment:
  Individual nodes: BARNUM
  Username SYSTEM will be used on nonlocal nodes

❸ SYSMAN> SET PROFILE/PRIVILEGE=CMEXEC
❹ SYSMAN> PARAMETERS SHOW /ACP
❺ %SYSMAN-I-USEACTNOD, a USE ACTIVE has been defaulted on node BARNUM
Node BARNUM: Parameters in use: ACTIVE
```

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
ACP_MULTIPLE	0	0	0	1	Boolean	D
ACP_SHARE	1	1	0	1	Boolean	
ACP_MAPCACHE	8	8	1	-1	Pages	D
ACP_HDRCACHE	36	128	3	-1	Pages	D
ACP_DIRCACHE	36	80	2	-1	Pages	D
ACP_DINDXCACHE	9	25	2	-1	Pages	D
ACP_WORKSET	0	0	0	-1	Pages	D
ACP_FIDCACHE	64	64	0	-1	File-Ids	D
ACP_EXTCACHE	64	64	0	-1	Extents	D
ACP_EXTLIMIT	100	100	0	1000	Percent/10	D
ACP_QUOCACHE	21	64	0	-	1 Users	D
ACP_SYSACC	4	8	0	-1	Directories	D
ACP_MAXREAD	32	32	1	64	Blocks	D
ACP_WINDOW	7	7	1	-1	Pointers	D
ACP_WRITEBACK	1	1	0	1	Boolean	D
ACP_DATACHECK	2	2	0	3	Bit-mask	D
ACP_BASEPRIO	8	8	4	31	Priority	D
ACP_SWAPFLGS	14	15	0	15	Bit-mask	D
ACP_XQP_RES	1	1	0	1	Boolean	
ACP_REBLDSYSD	1	1	0	1	Boolean	

Example Notes: Displaying System Parameters

Notes on the Displaying System Parameters example include:

- ❶ OPER privilege is needed to run SYSMAN.
- ❷ Establish the target node.
- ❸ Set the privilege needed to enable the next command.
- ❹ Display the value of a group of parameters. You can display a specific parameter.
- ❺ USE ACTIVE is assumed, meaning that the active parameters are displayed.
(Refer to the SYSGEN documentation for details on the use of USE ACTIVE, USE DEFAULT, and USE CURRENT.)

Maintaining the DECnet for OpenVMS Remote Node Database

Overview

Whenever someone adds a new node to the DECnet for OpenVMS network, or changes its name or address, you need to update the remote node database on your system.

The remote node database is part of the DECnet configuration database, which is maintained with a utility called the Network Control Program (NCP).

This section introduces the configuration database and NCP, and then describes the commands you use to update the remote node database to reflect changes in the network.

Topics

In this section the following topics are discussed:

- DECnet configuration database
- Using NCP
- Adding remote nodes
- Changing a remote node entry
- Removing remote nodes

DECnet Configuration Database

The DECnet configuration database contains data about the network. There are several different types of data in the database, including:

- **Nodes** — other systems connected to the network (this chapter covers the maintenance of node data)
- **Objects** — network applications on the local node
- **Circuits** — communication paths established between nodes by the DECnet software
- **Lines** — physical data paths between nodes

The DECnet configuration database consists of two distinct databases, one permanent and one volatile.

Every node in the network has its own DECnet configuration database. The information in the database is initialized when the DECnet software is first configured.

Permanent Database

The permanent database is a collection of files in SYSS\$SYSTEM that contain permanent settings for the network parameters.

Features of the permanent database include:

- It resides on disk.
- It provides initial values for the volatile database.
- Changes to the permanent database take effect the next time the DECnet software is started up.

Volatile Database

The volatile database is a memory-resident image containing current information about network management components.

Features of the volatile database include:

- It allows changes to be made while the DECnet software is running.
- Changes to the volatile database take effect immediately.
- Values in the volatile database are lost when the system is shut down.

Using NCP

NCP

The Network Control Program (NCP) utility provides commands to configure, control, monitor, and test a DECnet network.

Invoking NCP

To invoke NCP, enter the following command:

RUN SYS\$SYSTEM:NCP

Example:

```
$ RUN SYS$SYSTEM:NCP
NCP>
```

Exiting from NCP

To end an NCP session, enter EXIT or press Ctrl/Z.

```
NCP>EXIT
$
```

NCP Commands for Maintaining the Configuration Database

To maintain the DECnet configuration database, use these NCP commands:

Function	Command for Volatile Database	Command for Permanent Database
Create/modify parameters	SET	DEFINE
Delete parameters	CLEAR	PURGE
Display parameters	SHOW	LIST

Privileges Needed to Use NCP

The database is protected against tampering by nonprivileged users, so you must be logged in to a privileged account to access the permanent database or modify the volatile database. This table lists the privileges required for various NCP commands.

Command(s)	Privilege(s)	Reason
SHOW	None	SHOW does not affect either database, nor does it access the permanent database.
SET, CLEAR	OPER	These commands affect the volatile database only, but must still be used with caution.
LIST	SYSPRV	The LIST command displays information from the permanent database. Only users with system privilege can access the permanent database. To display password information, BYPASS privilege is needed as well.
DEFINE, PURGE	OPER, SYSPRV	These commands manipulate the permanent database. OPER is needed to manipulate any database; SYSPRV is needed to access the permanent database.

Protection on Database Files

To ensure that users cannot access the permanent database without SYSPRV privilege, the protection mask for the NET*.DAT files in SYS\$SYSTEM is set by default to:

- (S:RWED,O:RWED,G,W)

You may want to monitor these values regularly using the DIRECTORY /PROTECTION command to ensure that no tampering has occurred.

Adding Remote Nodes

Identifying Remote Nodes

Each node in a DECnet for OpenVMS (Phase IV) network is identified in your remote node database by two values:

- The node *name*, which consists of up to six alphanumeric characters
- The node *address*, in the form **aa.nnnn**, where:
 - **aa** is the area number (from 1 to 63, or 0 if the network is not divided into areas)
 - **nnnn** is the node number within the area (from 1 to 1024)

For each node, both the name and address must be entered in the remote node database. Otherwise, users cannot perform DECnet operations such as sending mail or transferring files to that node (unless they know the node's address).

Guaranteeing Unique Names and Addresses

Every node in the network should have a unique name and a unique address.

Because the DECnet software does not automatically detect conflicting names or addresses, someone must act as a **network administrator**, assigning names and addresses to physical nodes and guaranteeing that they are unique.

If the network is divided into multiple areas, the network administrator needs some knowledge of DECnet routing mechanisms.

Strategy for Setting Up Node Names

As network manager, you are allowed to assign some or all of the names related to clusters and networking. You can, if you like, assign a different name to each object in the system. However, such an approach might easily confuse your users.

Compaq recommends that you establish a system for assigning names. Assigning names in a systematic way will save you and your users time and trouble.

Example: Setting Up a Node Name

The following table shows how you might set up a node name.

Name for...	Example
DECnet/OSI full name	OMNI::US.Massachusetts.Boston.RUBY
DECnet/OSI node synonym	RUBY
DECnet for OpenVMS node name	RUBY
SCS node name	RUBY
LAT service name	RUBY
TCP/IP name	ruby.omni.com

Steps for Adding a Node

To add a node to the remote node database, follow these steps:

1. Obtain the node name and address from the network administrator or the system manager of that node.
2. Make sure that neither the name nor the address is currently defined in the remote node database. There are two ways to do this:

Enter the NCP command `SHOW KNOWN NODES` and examine the resulting list of names and addresses.

Enter the NCP command `SHOW NODE node-name` or `SHOW NODE address`.

If the name and address appear together in the output, the node is correctly defined and you need not continue with this procedure.

If the name appears with a different address, or the address appears with a different name, you must remove that information from the database.

3. Use the `DEFINE NODE` command to add the node to the permanent database. This command can take one of two forms:

`DEFINE NODE node-name ADDRESS address`

`DEFINE NODE address NAME node-name`

4. Use the `SET NODE ... ALL` command to copy the node information from the permanent database into the volatile database. This command can take one of two forms:

`SET NODE node-name ALL`

`SET NODE address ALL`

5. If you wish, enter a `SHOW KNOWN NODES` or `SHOW NODE` command to confirm that the node was added with the correct name and address.

Example: Adding a Remote Node

This example shows how to add a node named TOYS with address 3.4 to the remote node database.

❶ NCP>SHOW KNOWN NODES

Known Node Volatile Summary as of 20-NOV-1999 14:28:12

Executor node = 26.60 (LUIGI)

State = on

Identification = DECnet for OpenVMS V7.2

Node	State	Active Links	Delay	Circuit	Next node
4.20 (FIGMEN)				QNA-0	26.15 (CYCLPS)
24.29 (ANCHOR)				QNA-0	26.15 (CYCLPS)
26.15 (CYCLPS)				QNA-0	26.15 (CYCLPS)
26.61 (ENT)		1	1	QNA-0	26.15 (CYCLPS)
26.130 (SWSVAX)				QNA-0	26.15 (CYCLPS)
26.143 (PARROT)				QNA-0	26.15 (CYCLPS)
26.148 (TBD3)				QNA-0	26.15 (CYCLPS)
.					
.					
.					

❷ NCP>DEFINE NODE 3.4 NAME TOYS

❸ NCP>SET NODE 3.4 ALL

❹ NCP>SHOW KNOWN NODES

Known Node Volatile Summary as of 20-NOV-1999 14:30:22

Executor node = 26.60 (LUIGI)

State = on

Identification = DECnet for OpenVMS V7.2

Node	State	Active Links	Delay	Circuit	Next node
3.4 (TOYS)				QNA-0	26.15 (CYCLPS)
4.20 (FIGMEN)				QNA-0	26.15 (CYCLPS)
24.29 (ANCHOR)				QNA-0	26.15 (CYCLPS)
26.15 (CYCLPS)				QNA-0	26.15 (CYCLPS)
26.61 (ENT)		1	1	QNA-0	26.15 (CYCLPS)
26.130 (SWSVAX)				QNA-0	26.15 (CYCLPS)
26.143 (PARROT)				QNA-0	26.15 (CYCLPS)
26.148 (TBD3)				QNA-0	26.15 (CYCLPS)

.
.
.

Example Notes: Adding a Remote Node

Notes on the Adding a Remote Node example include:

- ❶ Make sure that neither the name nor the address is currently defined in the remote node database.
- ❷ Add the node to the permanent database.
- ❸ Copy the node information from the permanent database into the volatile database.
- ❹ Confirm that the node was added correctly.

Procedure for Copying Known Nodes

If you have many nodes to add to the remote node database, it may be cumbersome to enter `DEFINE NODE` and `SET NODE` commands for each node. An alternative is to copy the entire remote node database from another system by using the following procedure:

1. Add a node to the database that has a complete list of node definitions.
`NCP>SET NODE 2.4 NAME COOKIE`
2. To copy the remote node information from node `COOKIE` to both the permanent and volatile databases, use one of the following commands:

To preserve any node definitions that exist in the databases:
`NCP>COPY KNOWN NODES FROM COOKIE TO BOTH`

To empty the databases before repopulating them with information from a remote node:
`NCP>COPY KNOWN NODES FROM COOKIE TO BOTH WITH PURGE`
3. Use the `SHOW KNOWN NODES` or `LIST KNOWN NODES` command to ensure that the remote nodes are defined on your system.

Changing a Remote Node Entry

Reasons for Changing a Remote Node Entry

There are several reasons for changing a remote node entry:

- The node changes its address (perhaps because of a change in physical location, which may bring it into a different area).
- The node changes its name (perhaps to conform to a new naming convention, or perhaps to avoid conflicts when two networks are merged).
- The node name is misspelled or the address is entered incorrectly.

How to Change a Remote Node Entry

You can change a remote node's node name or node address by reentering the node in your database with the new name or address.

Example: Changing a Remote Node Entry

In this example, the address of node SPRUCE is changed from 26.89 to 4.118 in the remote node database. Before setting a new address for SPRUCE, two precautionary measures are taken:

1. Clearing the node SPRUCE, which removes the node from the volatile database
2. Purging the node SPRUCE, which removes the node from the permanent database

```
NCP>SHOW NODE SPRUCE
```

```
Node Volatile Summary as of 20-NOV-1999 14:32:12
```

Node	State	Active Links	Delay	Circuit	Next node
------	-------	--------------	-------	---------	-----------

26.89 (SPRUCE)			QNA-0		26.143 (PARROT)
----------------	--	--	-------	--	-----------------

```
NCP>
```

```
NCP>CLEAR NODE SPRUCE ALL
```

```
NCP>PURGE NODE SPRUCE ALL
```

```
%NCP-I-SUCCESS, Success
```

```
Remote node = 26.89 (SPRUCE)
```

```
%NML-I-RECDELET, Database entry deleted
```

```
NCP>SET NODE SPRUCE ADDRESS 4.118
```

```
NCP>SHOW NODE SPRUCE
```

```
Node Volatile Summary as of 20-NOV-1999 14:46:53
```

Node	State	Active	Delay	Circuit	Next node
------	-------	--------	-------	---------	-----------

Links

4.118 (SPRUCE)
NCP>

QNA-0

26.143 (PARROT)

Removing Remote Nodes

If a remote node is being permanently removed from the network, you should remove its name and address from your remote node database to prevent conflicts if its name or address is reused.

Procedure for Removing a Remote Node

To remove a remote node from the remote node database:

1. Make sure the node is in the database by using a `SHOW NODE` command.
2. Use the `CLEAR NODE` command to remove the node entry from the volatile database.
3. Use the `PURGE NODE` command to remove it from the permanent database.
4. You may want to enter a `SHOW NODE` command to confirm that the node is no longer in the volatile database, and a `LIST NODE` command to confirm that it is no longer in the permanent database.

Example: Removing a Remote Node

This example shows how to delete a node named ZODIAC from the remote node database.

❶ `NCP>SHOW NODE ZODIAC`

Node Volatile Summary as of 20-NOV-1999 14:45:43

Node	State	Active Links	Delay	Circuit	Next node
26.190 (ZODIAC)				QNA-0	26.143 (PARROT)

NCP>

❷ `NCP>CLEAR NODE ZODIAC ALL`

❸ `NCP>PURGE NODE ZODIAC ALL`

❹ `NCP>SHOW NODE ZODIAC`

Node Volatile Summary as of 20-NOV-1999 14:47:33

%NCP-W-UNRCMP, Unrecognized component , Node

Example Notes: Removing a Remote Node

Notes on the Removing a Remote Node example include:

- ❶ Make sure the node is in the database.
- ❷ Remove the node entry from the volatile database.
- ❸ Remove the node entry from the permanent database.
- ❹ Confirm that the node is no longer in the volatile database.

Summary

Concepts

Customizing System Startup Files

- The site-independent system startup file, `STARTUP.COM`, is responsible for:
 - Starting up other system startup files that must be run
 - Executing autoconfiguration commands, unless canceled by the user
 - Installing optional swap files
- Starting the configuration process, if the parameter is set accordingly
- `SYSS$MANAGER:SYCONFIG.COM` is used to connect special devices to the system and to load their I/O drivers.
- `SYSS$MANAGER:SYLOGICALS.COM` is used to define systemwide logical names.
- `SYSS$MANAGER:SYPAAGSWPFILES.COM` is used to install page and swap files on disks other than the system disk.
- `SYSS$MANAGER:SYSTARTUP_VMS.COM` is the site-specific command procedure invoked by `STARTUP.COM`.

Customizing Login Command Procedures

- The OpenVMS operating system allows you to specify the user's environment using login command procedures.
- Login command procedures can be user specific, group specific, systemwide, and other.
- User-specific login command procedures define the environment for each individual user.
- Systemwide login command procedures define the environment for all users.
- Group login command procedures define the environment for a group of users.
- Other login command procedures, such as captive command procedures, are created by the system manager or the user.

Customizing System Parameters

- The system manager can modify system parameters to control such things as:
 - Sizes of OpenVMS data structures in memory
 - Number of installed image files
 - Sizes of system files
- The OpenVMS operating system provides a variety of tools to modify system parameters.
- The AUTOGEN command procedure is the recommended method for changing system parameters and file sizes.
- Use the SYSMAN utility to maintain system parameters in more than one computer system.

Maintaining the DECnet for OpenVMS Remote Node Database

- There are two DECnet configuration databases:
 - Permanent
 - Volatile
- The Network Control Program (NCP) is used to manipulate DECnet databases.
- When a node is added to the network, the information in the remote node database must be modified.
- When the characteristics of a node are changed, the information in the remote node database must be modified.
- When a node is removed from the network, the information in the remote node database must be modified.

Commands

Customizing System Startup Files

AUTOCONFIGURE ALL

On VAX systems, SYSGEN command to connect and configure nonstandard devices.

IO CONNECT

On Alpha systems, SYSMAN command to connect and configure nonstandard devices.

ASSIGN/SYSTEM[/EXECUTIVE]

Creates a systemwide logical name, optionally in executive mode.

DEFINE/SYSTEM[/EXECUTIVE]

Creates a systemwide logical name, optionally in executive mode.

DEASSIGN/SYSTEM

Deletes a systemwide logical name.

INSTALL file-specification/PAGEFILE

INSTALL file-specification/SWAPFILE

SYSGEN commands to install page and swap files.

Customizing Login Command Procedures

MODIFY user /LGICMD=file-specification

AUTHORIZE command that specifies a login command procedure for a user.

Customizing System Parameters

@SYSS\$UPDATE:AUTOGEN [start-phase] [end-phase] [mode]

Invokes the AUTOGEN command procedure.

PARAMETERS SHOW parameter-specification

SYSMAN command to display the value of a parameter or a group of parameters.

Maintaining the DECnet for OpenVMS Remote Node Database

RUN SYSSYSTEM:NCP

Invokes NCP.

SHOW KNOWN NODES

NCP command to display all nodes in the volatile database.

LIST KNOWN NODES

NCP command to display all nodes in the permanent database.

SHOW NODE node-name

NCP command to display a single node in the volatile database.

DEFINE NODE address NAME node-name

NCP command to register a node address and name in the permanent database.

SET NODE node-name ALL

NCP command to update the volatile database for a node.

CLEAR NODE node-name ALL

NCP command to remove all information for a node from the volatile database.

PURGE NODE node-name ALL

NCP command to remove all information for a node from the permanent database.

Installing Layered Products

Module 6

About This Module

Introduction

A variety of optional (layered) software products are available to run under the control of the OpenVMS operating system. To use these products you must purchase a license that comes in the form of a product authorization key (PAK).

Install layered software products after installing the OpenVMS operating system and running the User Environment Test Package (UETP). Use `SYSS$UPDATE:VMSLICENSE.COM` to register the PAK for the product.

To install layered software products on the system disk, you use either the `VMSINSTAL` command procedure or the `POLYCENTER` Software Installation utility (if supported by the particular layered product). To complete the installation of a layered software product, you might also need to modify limits in UAF records, create new directories and logical names, change system parameters, as well as other tasks.

Some layered software products are part of the OpenVMS kit and require no further installation. These **system integrated products** include:

- DECnet for OpenVMS
- OpenVMS Cluster software
- Volume Shadowing for OpenVMS
- OpenVMS RMS Journaling

Some layered software products must be installed in a certain order. Read the product's installation guide to find out what the order is.



Note

The complete DECwindows product is no longer included (bundled) with the OpenVMS operating system. During an installation or upgrade of the operating system, system managers can install DECwindows base and workstation support, but after the installation is complete, they must also install the separate DECwindows for OpenVMS layered product to obtain full DECwindows support.

See the DECwindows for OpenVMS installation documentation for more information.

This module presents the basic concepts used in installing layered software products. Topics covered include:

- The responsibilities of managing product licenses
- The steps used in installing layered software products

**Note**

Alpha Differences Some tasks are different or new on OpenVMS Alpha. Differences are in the following areas:

- Use of the License Management facility (LMF)
 - Product authorization keys (PAKs)
 - VMSINSTAL utility
-

Objectives

To install layered software products, you should be able to:

- Describe the License Management facility (LMF).
- Use the License Management utility (LICENSE) to manage the software license database.
- Install layered software products on a single OpenVMS system or in an OpenVMS Cluster environment.

Managing Product Licenses

Overview

Many Digital software products, including the OpenVMS operating system, require that a product license be registered and activated in the license database, a new database provided specifically for license information.

Install the license by copying information from a paper **product authorization key (PAK)**. You enter product licenses into a clusterwide database.

As each node is started up, licenses are loaded into a license database in memory.

- Products on any node in the cluster check the license database to determine whether a particular use of the product is licensed or not.
- There are three basic types of licenses:
 - **Availability license** — Allows the use of a product on a host
 - **Activity license** — Allows a specific number of concurrent users to access a product
 - **Personal use license** — Allows a specific number of named users unlimited access to a product

Products can also be grouped together under a single license so that a single license will enable the use of several products.

Topics

In this section the following topics are discussed:

- License Management facility (LMF)
- License Management utility (LICENSE)

The License Management Facility

The License Management Facility (LMF) provides a variety of system-level software components used to maintain software license information in the LICENSE database.

Components of the LMF

Components of the LMF include:

- LICENSE database
 - Holds all information about licenses. The database is managed by the LICENSE utility.
 - SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB is the default location.
 - If there is more than one system disk in the cluster, the logical name LMF\$LICENSE should point to a single database for the cluster containing all product licenses for all cluster hosts. This can be defined in SYLOGICAL.COM.
 - If there is no disk accessible to all hosts, separate databases should be maintained **identically**.
- License management utility (LICENSE)
 - DCL-level interface to the LMF on the OpenVMS operating system.
- SYS\$UPDATE:VMSLICENSE.COM
 - A command procedure to assist you in registering your licenses in the license database.
 - Avoids the cumbersome, error-prone typing of the LICENSE REGISTER command.
 - Can be used in place of the LICENSE LIST and LICENSE MODIFY commands.
 - Entered data is retained to be used as default data during the next PAK registration or license management task.
 - Task can be specified on the DCL command line to bypass first menu in the VMSLICENSE procedure (@SYS\$UPDATE:VMSLICENSE.COM REGISTER).
- Can register PAKs using batch processing. Data file is entered as a parameter when you invoke the COM file.

**Note**

Stress at this point that the LICENSE database is used ONLY when the product (layered or otherwise) makes an inquiry. Most people have the impression that the LMF can somehow stop software from running. THIS IS NOT TRUE. The software asks for information about what is in the database. The software, not the LMF, makes the decision about running or not. If the product (whether produced by Digital or another vendor) does not query the database, LMF does NOT stand in the way of the product running.

License Units

The following list defines the concept of license unit:

- Element that specifies how much product use a license authorizes.
- Number of individual units purchased with any license is specified by the PAK.
- Each processor has a series of license unit requirements.
- Products query the LMF to determine if there are sufficient units available to activate the product.

**Note**

Features of the **license unit** that provide ways to move software:

- Basic element by which Digital specifies how much product use a license authorizes.
 - Each license key specifies the number of units purchased with the corresponding license.
 - Each processor has a series of license unit requirements, with higher-performance processors requiring more license units.
 - Products query the LMF to determine if there are sufficient units available to activate the product. If the number of available units is greater or equal to the units required, then the product may be used.
-

Activity Licenses

An activity license determines the number of concurrent users allowed for a product at one time.

- The user can be anywhere in the cluster.
- Different hosts may require different numbers of units per user. If the PAK lists a value in the Activity field, the unit requirement is that value and is the same for all users on all nodes.
- As each user activates the product, the number of available units decreases.
- If there are not enough available units, that user gets an error message.
- When a user stops using the product, the units they were using become available.
- Use the SHOW LICENSE/USAGE command to monitor activity license usage by users.



Note

- Activity licenses are useful when you expect to use a product infrequently and by a small number of users on a large machine.
 - Additional activity licenses can be purchased to add more users.
 - Products licensed by activity licenses can also be licensed by availability licenses at your discretion if product usage increases beyond the point where an activity license is cost effective.
-

Product Usage with Activity License in a Cluster

As an example, say a product called QUERY has an activity license for 1000 units in a cluster consisting of:

- VAX 8650
- VAX 8350
- Three VAXstation 2000 systems

As each user anywhere in the cluster accesses QUERY, the number of units required per user by that host will be subtracted from the number of units available to all other hosts in the cluster.

- Eight users on the VAX 8650 require $8 \times 75 = 600$ units.
- Five users on the VAX 8350 at the same time require an additional $5 \times 60 = 300$ units.
- This leaves only 100 units.
- Allows each of the three VAXstation 2000s to have one user of QUERY.
- The next user anywhere in the cluster would receive an error message.

License Management Utility

LICENSE is a DCL-level interface to the license management facility (LMF) on the OpenVMS operating system.

Entering LICENSE Commands

To use the license management utility (LICENSE), enter the LICENSE command and the desired LICENSE subcommand and qualifiers at the DCL prompt (\$).

Format:

LICENSE subcommand parameter

You can enter short commands on one line:

```
$ LICENSE LIST VAX-VMS/AUTHORIZATION=USA12345
```

Or, you can enter long command strings followed by a hyphen (-) before pressing Return:

```
$ LICENSE REGISTER FORTRAN /ISSUER=DEC -  
_ $ /AUTHORIZATION=USA-1999 -  
_ $ /PRODUCER=DEC /UNITS=1200 /VERSION=6.0 -  
_ $ /AVAILABILITY=F /CHECKSUM=1-GEAD-MIDJ-IDNC-DLAC
```



Note

This is not correct license information, see your instructor to obtain a working license if you need one.

Because LMF may return only a checksum error when you omit or incorrectly enter information, carefully check the characters that you entered before pressing Return.

The command procedure SYS\$UPDATE:VMSLICENSE.COM eliminates much of the typing needed for the LICENSE REGISTER and LICENSE AMEND commands.

Privileges Needed to Use LICENSE

Most LICENSE commands require only the privileges needed to access the license database.

- Normal OpenVMS file protection applies.
- The LMF provides the database with a default file access of read and write privileges to system-level processes (S:RW).

The LICENSE START, LICENSE LOAD, and LICENSE UNLOAD commands need the following privileges:

- CMKRNL
- SYSNAM
- SYSPRV

Listing Licenses

To display a listing of licenses registered in the license database, use the LICENSE LIST command.

To display the licenses active on the current node, use the DCL command SHOW LICENSE. For OpenVMS VAX Version 6.0, improvements have been made to the SHOW LICENSE command to make the command consistent with the License Management utility's LICENSE LIST command. The SHOW LICENSE command:

- Lists names attached to a license (known as a RESERVE list)
- Displays the new PAK option, RESERVE_UNITS
- Lists all LURT columns that are loaded

Loading and Unloading Licenses

The LICENSE LOAD and LICENSE UNLOAD commands let you load and unload licenses clusterwide.

```
SYSMAN> LICENSE LOAD product [/DATABASE=filespec] [/PRODUCER=string]  
SYSMAN> LICENSE UNLOAD [product] [/PRODUCER=string]
```

Enabling Licenses

When you enter a LICENSE ISSUE command, LMF disables the license in the current license database and marks the license disabled. To enable a license that has been marked DISABLED, enter the LICENSE ENABLE command.

Sample Product Authorization Key

The following shows an example of a Product Authorization Key:

PAK ID:

 Issuer: DEC
 Authorization Number: USA000877

PRODUCT ID:

 Product Name: FORTRAN
 Producer: DEC

NUMBER OF UNITS:

 Number of units: 5000

KEY LEVEL:

 Version: 6.0
 Product Release Date: 18-FEB-1999

KEY TERMINATION DATE:

 Key Termination Date:

RATING:

 Availability Table Code: F
 Activity Table Code:

MISCELLANEOUS:

 Key Options: MOD_UNITS
 Product Token:
 Hardware-Id:
 Checksum: 1-CCLB-MNBO-KNNG-CBEH



Note

This is not correct license information. See your instructor to obtain a working license if you need one.

Example: VMSLICENSE Session

The following is an example of a session using VMSLICENSE:

```
$ SET DEFAULT SYS$UPDATE
$ @VMSLICENSE.COM
```

VMS License Management Utility Options:

1. REGISTER a Product Authorization Key
2. AMEND an existing Product Authorization Key
3. CANCEL an existing Product Authorization Key
4. LIST the Product Authorization Keys
5. MODIFY an existing Product Authorization Key
6. DISABLE an existing Product Authorization Key
7. DELETE an existing Product Authorization Key
8. COPY an existing Product Authorization Key
9. MOVE an existing Product Authorization Key
10. ENABLE an existing Product Authorization Key
11. SHOW the licenses loaded on this node
12. SHOW the unit requirements for this node

99. EXIT this procedure

Type '?' at any prompt for a description of the information requested. Press Ctrl/Z at any prompt to exit this procedure.

Enter one of the above choices [1]: 1

Do you have your Product Authorization Key? [YES]: y

Use the REGISTER option to add a new license to a license database. A Product Authorization Key (PAK) provides the product name and information you need to register the license. You must enter all the information provided by your PAK exactly as it appears.

Type '?' at any prompt for a description of the information requested. Press Ctrl/Z at any prompt to return to the main menu.

```
      Issuer [DEC]: [Return]
    Authorization Number []: USA000877
      Product Name []: FORTRAN
      Producer [DEC]: [Return]
    Number of Units []: 5000
      Version []: 6.0
    Key Termination Date []: [Return]
Availability Table Code []: F
  Activity Table Code []: [Return]
      Key Options []: MOD_UNITS
    Product Token []: [Return]
    Hardware-Id []: [Return]
      Checksum []: 1-CCLB-MNBO-KNNG-CBEH
```

Here is a list of the license information just entered:

Issuer: DEC
Authorization: USA000877
Product Name: FORTRAN
Producer: DEC
Units: 5000
Release Date:
Version: 6.0
Termination Date:
Availability: F
Activity:
Options: MOD_UNITS
Token:
Hardware ID:
Checksum: 1-CCLB-MNBO-KNNG-CBEH

Is that correct? [YES]:

Registering FORTRAN license in SYS\$COMMON:[SYSEXE]LMF\$LICENSE.LDB...

Do you want to LOAD this license on this system? [YES]: [Return]

%LICENSE-I-LOADED, DEC FORTRAN was successfully loaded with 5000 units

VMS License Management Utility Options:

1. Register a Product Authorization Key

.
. <other options omitted from this listing>
.

Enter one of the above choices [1]: 99

\$

License Management Facility and License Agreements

The terms and conditions of your product contract determine your legal use of software. The LMF is a management tool that can help you comply with your license agreement. However, the LMF offers options for many kinds of license agreements. Using some of these options can be illegal for your specific contract. You must read your contract carefully to determine which LMF options you can legally use.

For more information on your legal responsibilities, contact your Digital representatives.

Hints on Running VMSSLICENSE

- If the version number is omitted on the PAK, do not add it when answering the questions.
- If you are repeating the session, the answers that you gave before will be filled in as the defaults.
- If you need to clear a field, use the backslash (\).

Installing Layered Products

Overview

For a user to be granted access to a layered software product, you may need to modify their account privileges or quotas. Some layered software products may recommend making changes to system parameters using the SYSGEN utility. The OpenVMS operating system provides two methods for installing layered products: the VMSINSTAL command procedure and the POLYCENTER Software Installation utility. Installation of layered software products on a common system disk requires special consideration.

Topics

In this section the following topics are discussed:

- Adjusting user privileges and quotas
- Adjusting system parameters
- Installing layered products on a common system disk
- Installations using VMSINSTAL
- Installations using the POLYCENTER Software Installation utility

Adjusting User Privileges and Quotas

Privileges and Quotas

For a user to use a layered product, you may need to modify some user account privileges or quotas. User account privileges and quotas are stored in the file SYSUAF.DAT.

How to Adjust User Privileges and Quotas

Use the Authorize utility to verify and change user account privileges and quotas. To use the Authorize utility, set your default directory to SYS\$SYSTEM and enter the DCL RUN AUTHORIZE command as follows:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF>
```

To check a particular account, enter the SHOW command with an account name.

For example:

```
UAF> SHOW USER1
```

To change a privilege or quota, enter the MODIFY command in the following format:

MODIFY user-name/PRIVILEGE=privilege/quota-name=n

The following command adds the NETMBX privilege and modifies the BYTLIM quota for the USER1 account.

```
UAF> MODIFY SMITH /PRIVILEGE=NETMBX /BYTLIM=16384
```

After you exit the utility, the OpenVMS operating system displays messages indicating whether or not changes were made. Once you have made the changes, users must log out and log in again for the new privileges and quotas to take effect.

DEFAULT Account

Whenever a new account is created, its characteristics come from the DEFAULT account except for the ones that you specify. To change a limit or quota for any future accounts, modify them in the DEFAULT account that exists in every SYSUAF.DAT file.

Quotas and Resource Limits

There are many quotas and resource limits associated with a process. When a user logs in to an account, the created process is given the quotas and other limits defined in the account's UAF record.

You should always refer to the installation guide for the product that you are installing before beginning the installation. The guide may recommend making changes to either user quotas or system parameters.

AUTHORIZE Qualifiers for Quota Fields

The following table lists the AUTHORIZE qualifiers for the quota fields and other process-related parameters.

Qualifier	Function
/ASTLM=value	Number of ASTs the user can have queued at any one time.
/BIOLM=value	Maximum number of buffered I/O operations the user can have outstanding at any one time.
/BYTLM=value	Maximum number of bytes of nonpaged system dynamic memory that the user's job may consume at any one time.
/CPUTIME=time	Maximum CPU time a user's process can take per session, specified as a delta-time value.
/DIOLM=value	Maximum number of direct I/O operations (usually disk) that the user can have outstanding at any one time.
/ENQLM=value	Maximum number of locks that can be queued at any one time.
/FILLM=value	Maximum number of files that can be open at one time.
/JTQUOTA=value	Initial maximum number of bytes with which the jobwide logical name table is to be created.
/MAXACCTJOBS=value	Maximum number of batch, interactive, and detached processes that may be active at any one time for all users of the account. The default value of 0 represents an unlimited number.
/MAXDETACH=value	Maximum number of detached processes allowed at any one time.
/MAXJOBS=value	Maximum number of batch, interactive, detached, and network processes that may be active at any one time.
/PGFLQUOTA=value	Maximum number of pages the user's process can use in the system paging files.
/PRCLM=value	Maximum number of subprocesses that can exist at one time for the user's process.
/PRIORITY=value	Default base priority for all processes created by the user.
/SHRFILLM=value	Maximum number of shared files the user may have open at any one time.
/TQELM=value	Total number of entries in the timer queue, plus the number of temporary common event flag clusters the user can have at any one time.
/WSDEFAULT=value	Number of pages in the user's default working set.
/WSEXTENT=value	Number of pages in the user's working set extent.
/WSQUOTA=value	Number of pages in the user's working set quota.

Adjusting System Parameters

Some layered product installation guides specify requirements for system parameters. You can use the SYSGEN utility to adjust the values of the specified parameters.

Ensure Number of GBLPAGES and GBLSECTIONS

Some installation guides may specify that you must ensure that you have a certain number of GBLPAGES (**global pages**) and GBLSECTIONS (**global sections**) on your system. Adjust these as follows:

1. Invoke the Install utility:
\$ INSTALL
2. Display the existing global sections already known to the operating system:
INSTALL> LIST/GLOBAL/SUMMARY

The Install utility lists the following:

Number of global sections used

Number of global pages used and unused
3. Press Ctrl/Z to exit the Install utility.

Example: Output from the LIST COMMAND

The following example shows the commands that you would use to display the existing global sections and global pages:

```
$ INSTALL
INSTALL> LIST/GLOBAL/SUMMARY

Summary of Local Memory Global Sections
279 Global Sections Used, 23722/278 Global Pages
Used/Unused
```

```
INSTALL> [Ctrl/Z]
```

Use the SYSGEN utility to display the existing number of available global sections and global pages:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SHOW GBLSECTIONS
Parameter Name      Current  Default  Min.    Max.    Unit    Dynamic
-----
GBLSECTIONS         420     250     60    4095    Sections
SYSGEN> SHOW GBLPAGES
Parameter Name      Current  Default  Min.    Max.    Unit    Dynamic
-----
GBLPAGES            62000   15000    512   4194176 Pages
SYSGEN> EXIT
```


The system responds with data on global sections and global pages. The current maximum number of global sections and pages is the first number given in the respective system response.

Example: SYSGEN SHOW Command

The following example shows the commands that you would use to display the available global sections and pages.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE CURRENT
SYSGEN> SHOW GBLSECTIONS
Parameter Name  Current  Default  Min.    Max.    Unit    Dynamic
-----
GBLSECTIONS      340      250      60     4095    Sections
SYSGEN> SHOW GBLPAGES
Parameter Name  Current  Default  Min.    Max.    Unit    Dynamic
-----
GBLPAGES        24000    10000    512     -1      Pages
SYSGEN> EXIT
```

Example Notes: SYSGEN SHOW Command

Compare these numbers with the numbers currently used, as shown by the LIST command of the Install utility. If fewer than the required number of global pages or global sections remain, you should increase the number of available global sections or pages as follows:

- Edit the file SYS\$SYSTEM:MODPARAMS.DAT by adding or modifying the applicable lines indicated below (n and m are, respectively, the total number of global sections and global pages required for the new images added):
- ADD_GBLSECTIONS=n
- ADD_GBLPAGES=m
- Invoke the AUTOGEN utility to update the GBLSECTIONS and GBLPAGES parameters:

```
$ @SYS$UPDATE:AUTOGEN GETDATA REBOOT
```



Note

If a user receives the error message "SYSTEM-F-EXQUOTA, Quota Exceeded" error message, consider increasing quotas for the user, particularly PGFLQUOTA or BYTLM.

Commonly Changed System Parameters

The following table lists the most commonly changed system parameters for layered products.

This parameter...	Performs this function:
GBLPAGES	Establishes the size of the global page table and the limit for the total number of global pages that can be created
GBLSECTIONS	Determines the maximum number of global sections that can be made known to the system by allocating the necessary storage for the global section table (GST)
VIRTUALPAGECNT	Determines the total number of pages that can be mapped for a process, which can be divided in any fashion between P0 and P1 space

Installing Layered Products on a Common System Disk

Things to Know Before Installing a Product

Most installed products go to the common area of the disk by default.

Things to know before installing a product:

- Does the product:
 - Run under the current version of the operating system?
 - Support a cluster environment?
 - Require a PAK?
- Where to get the answers to these questions:
 - Software Product Description (SPD) for the product
 - Installation guide and release notes for the product
 - Digital sales or software representative

You can find system and UAF parameter settings in the SPD or the release notes.

Considerations When Installing Products on an OpenVMS Cluster System

The standard precautions for product installation apply for OpenVMS Cluster systems. For most products, the only concerns that may require more careful attention are system parameters and licensing.

- Install layered products as in a nonclustered environment.
- Perform the actual installation (the documented procedure) once for each system disk.
- Edit MODPARAMS.DAT to add the required number of global pages and sections on each node in the cluster.

When to Run AUTOGEN

It is recommended that you make the appropriate changes to MODPARAMS.DAT and run AUTOGEN before installing a layered product.

In a cluster, running AUTOGEN on each system would be time consuming. An efficient way to run it in the cluster would be to run AUTOGEN detached.

The following example shows one way to run AUTOGEN detached in a cluster:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
SYSMAN> DO
RUN/DETACH/INPUT=CL_AUTOGEN.COM/OUTPUT=CL_AUTOGEN.LOG -
_SYSMAN> /PROCESS=CL_AUTOGEN SYS$SYSTEM:LOGINOUT.EXE
SYSMAN> EXIT
$
$ TYPE CL_AUTOGEN.COM
$! CL_AUTOGEN.COM
$!
$ @SYS$UPDATE:AUTOGEN GETDATA SETPARAMS NOFEEDBACK
$!
```

After Installation

After installation, you should:

- Create product-specific files in the SYS\$SPECIFIC directory on each node, if necessary. VMSINSTAL informs you if you have to create a directory in SYS\$SPECIFIC.
- Modify any files in SYS\$SPECIFIC on each node that the procedure told you to modify.
- Reboot each node to ensure that:
 - The node is set up to run the product correctly.
 - The node is running the latest version of the product.
- Manually run the installation verification procedure (IVP), if you did not run it during installation.
 - Run it from at least one other node in the cluster, preferably from all nodes.
 - If VMSINSTAL deletes the IVP, you may be able to restore it with the following command procedure:

```
$ @VMSINSTAL product source_device OPTIONS G device:[directory]
$ BACKUP device:[directory]product.A/SELECT=KITINSTAL.COM device:[directory]
$ @device:[directory]:KITINSTAL VMI_IVP
```

Do not use search lists when creating files.

- Use SYSS\$SPECIFIC or SYSS\$COMMON instead of SYSS\$SYSROOT.
- Use SYSS\$SPECIFIC:[SYSEXE] or SYSS\$COMMON:[SYSEXE] instead of SYSS\$SYSTEM.

Summary of the VMSINSTAL Options

The following table lists the VMSINSTAL options and their functions.

This option...	Performs this function:
A	The auto-answer option makes it easier to reinstall a product by providing responses to VMSINSTAL questions and prompts during the reinstallation. The auto-answer option is used most often for reinstalling products after a system is upgraded.
AWD	The alternate working device option lets you specify an alternate working device for the temporary working directory. This option enables you to perform an installation with fewer free blocks on the VMI\$ROOT device than is otherwise required.
G	Use the GET save set option to copy the product kit save sets into a disk directory or other storage device for later installation. When you select option G, all kit save sets are copied, but no installation is performed.
L	Use the file log option to log all file activity to the controlling terminal during installation. File activity is defined as any action that alters the disposition of a file, such as creating a new file, updating a library, or deleting a file.
N	Use the release notes option to display or print the release notes file supplied by the layered product.
R	Use the alternate root option to install the product in a system root other than that of the running system. This option makes it possible to test a new product without disturbing the running system.

Installations Using the POLYCENTER (PCSI) Software Installation Utility

POLYCENTER Software Installation Utility

The POLYCENTER Software Installation (PSCI) utility offers rapid installation of layered products. In addition, this utility does the following:

- Lets you install many products at the same time (that is, with the same command)
- Lets you remove software products that have been installed using the POLYCENTER Software Installation utility
- Provides information about software products

You can use the POLYCENTER Software Installation utility to install those software products that conform to the utility's installation requirements. A software product that conforms to these requirements includes an explicit statement in its installation information to indicate that it can be installed using the POLYCENTER Software Installation utility.

You can use the POLYCENTER Software Installation utility with either a DECwindows Motif interface or a DCL keyboard user interface.



Note

The PRODUCT command can cause indexed and relative files to be truncated during a package or install operation. See the release notes for a workaround.

If you manually delete a directory created by the POLYCENTER Software Installation utility, a reinstallation of the same product will fail because it will not recreate the directory before attempting to write to it.

The POLYCENTER Software Installation utility does not check for disk quotas and sometimes underestimates the required disk space, so an installation may fail without prior warning.

Removing a product with the POLYCENTER Software Installation utility removes the accounts and rights identifiers associated with the product, without checking whether the product is still available on another system using the same SYSUAF.DAT and RIGHTSLIST.DAT.

Privilege Needed for the POLYCENTER Software Installation Utility

To use the POLYCENTER Software Installation utility, you must have the SYSPRV privilege on the local node and authorization for the SYSPRV privilege on any remote node on which you want to use this utility.

Using the DECwindows Motif Interface

To install one or more layered products with the POLYCENTER Software Installation utility using the DECwindows Motif interface, do the following:

1. Enter the PRODUCT command at the DCL prompt (\$), as follows:
\$ PRODUCT [Return]
The main window is displayed.
2. Pull down the Mode menu, and click on Installation.
The Install Product List window is displayed.
3. To install only one product, double click on the product name or the button to the left of that product.
To install more than one product, do the following:
 - a. Be sure that the Multiple Select Hold button in the upper right portion of the screen is filled in. (If the button is clear, click on it once to fill it in.)
 - b. Double click on each of the products that you want to install, scrolling as needed to find the products. As you choose each one, the diamond to the left of the product name is partially filled in.
 - c. After you have chosen all the products to install, clear the Multiple Select Hold button by clicking on it.
4. When the Selected Products window displays the products that you have chosen for installation, click on the Install button at the bottom of the screen. The POLYCENTER Software Installation utility then installs those products.

Using the DCL Keyboard Interface

To install one or more layered products with the POLYCENTER Software Installation utility using the DCL keyboard interface, do the following:

1. Verify that the logical name PCSI\$SOURCE points to the correct location for the products that you want to install, for example SYS\$UPDATE.
2. Obtain a list of the product kits that are available for installation on your system by entering the following command:

```
$ PRODUCT FIND *
```

3. Enter the PRODUCT INSTALL command at the DCL prompt. For example, to install COBOL Version 5.0 and LSE Version 3.0, enter the following command:

```
$ PRODUCT INSTALL COBOL/VERSION=5.0,LSE/VERSION=3.0 [Return]
```

The following products have been selected:

```
DEC OpenVMS COBOL V5.0
```

```
DEC OpenVMS LSE V3.0
```

Do you want to continue [YES]) [Return]

```
.  
.   
.
```


Summary

Concepts

Managing Product Licenses

- Many Digital layered products, including the OpenVMS operating system, require a license to be loaded on the system before they can be used.
- There are three types of licenses:
 - Availability — Allows use of a product on a host
 - Activity — Allows a specific number of concurrent users to access the product
 - Personal use — Allows a specific number of named users unlimited access to the product
- Use The License Management Facility (LMF) to maintain license information on an OpenVMS system.
- Components of the LMF include:
 - The license database
 - VMSLICENSE.COM
- The License Management utility is a command-line interface to the LMF. The utility allows you to:
 - List all licenses or single licenses
 - Register product authorization keys (PAKs)
 - Modify licenses
 - Delete licenses
 - Show all loaded licenses

Installing Layered Products

- When installing a layered product, you may need to adjust system parameters such as:
 - Global pages (GBLPAGES)
 - Global section (GBLSECTIONS)
- In a clustered environment, the system manager must decide whether to install layered products in the cluster common directories.
- VMSINSTAL is a command procedure used to install layered products.
- The POLYCENTER Software Installation utility uses a DECwindows Motif interface or a DCL keyboard user interface to install layered products.

Commands

Managing Product Licenses

@SYSS\$UPDATE:VMSLICENSE

Used to register licenses in the license database.

LICENSE

DCL interface to the License Management Facility.

Installing Layered Products

AUTHORIZE

Used to check account quotas.

INSTALL LIST/GLOBAL/SUMMARY

Displays the number of global sections used, and the number of global pages used and unused.

SHOW GBLSECTIONS

SYSGEN command to display the total number of global sections available.

@SYSS\$UPDATE:AUTOGEN

DCL command to invoke the AUTOGEN utility to automatically adjust system parameters.

PRODUCT

DCL command to invoke the POLYCENTER Software Installation utility.

Monitoring System Resources

Module 7

About This Module

Introduction

The OpenVMS operating system provides many tools for monitoring system behavior. Using these tools, you can examine the activity of the OpenVMS processes and the utilization of system resources.

At first, you use these tools to become familiar with the characteristics of your system when it is functioning normally. If you ever observe differences from these characteristics, a problem may be occurring, and someone experienced in OpenVMS support can solve the problem using the information displayed by these tools.

As you become more experienced in system management and start to perform troubleshooting and system performance management, these same tools provide you with the information you need to perform those tasks.

This module introduces some commonly used tools and some of their features.

Objectives

To monitor system activity, you should be able to:

- Monitor the OpenVMS Cluster system by using the various SHOW CLUSTER commands.
- Use the appropriate DCL commands to help monitor system resource usage.
- Monitor a network using the basic network control program (NCP) commands.

Using the Show Cluster Utility

Overview

The Show Cluster utility displays a variety of information about an OpenVMS Cluster system. Much of this information is not intelligible without some knowledge of OpenVMS Cluster software internals, but there is some basic information that is useful for OpenVMS Cluster system management.

Topics

In this section the following topics are discussed:

- Show Cluster utility
- Tailoring the display
- Using an initialization file

Show Cluster Utility

The Show Cluster utility monitors nodes in an OpenVMS Cluster system and displays information about activity and performance. You can display this information at your terminal or store it in a file.

Running SHOW CLUSTER

There are two ways to run SHOW CLUSTER, yielding two different types of displays.

- SHOW CLUSTER yields a one-time display.
- SHOW CLUSTER/CONTINUOUS yields a dynamic display that is continuously updated.

In either case, SHOW CLUSTER provides a view of the cluster as seen from a single node, rather than a complete view of the cluster.



Note

Most of the fields in SHOW CLUSTER are not useful until the student learns something about OpenVMS Cluster software internals. Stick to the few easily understood fields shown here.

Example: SHOW CLUSTER Default Display

By default, the display produced by SHOW CLUSTER looks like this:

```
$ SHOW CLUSTER
```

```
View of Cluster from system ID 62838 node: NEAT 23-SEP-1999
10:01:13
```

```
+-----+-----+
| SYSTEMS      | MEMBERS      |
+-----+-----+-----+
| NODE         | SOFTWARE     | STATUS      |
+-----+-----+-----+
| NEAT         | 7.2          | MEMBER      |
| TIDY         | 7.2          | MEMBER      |
| SLOPPY       | 7.2          | MEMBER      |
| MESSY        | HSC V400    |              |
+-----+-----+-----+
```

Example Notes: SHOW CLUSTER Default Display

The display shows these fields, in order from left to right:

- **NODE** — a list of nodes currently in the cluster. VAX, Alpha, HSC, and ISE nodes are all included.
- **SOFTWARE** — the operating system, including the version number, that the node is running.
- **STATUS** — whether the node is a cluster member. Nodes that run the OpenVMS operating system are members. Device controllers, HSC, and ISE nodes are not members.



Note

SYSTEMS and MEMBERS are classes of information.

- **NODE** and **SOFTWARE** are fields within the **SYSTEMS** class.
- **STATUS** is a field within the **MEMBERS** class.

Additional SHOW CLUSTER Display Fields

Here are some additional fields that you might want to add to the SHOW CLUSTER display. The documentation lists many others.

This field...	Contains this information:
TRANSITION_TIME	The time when each node most recently booted (joined the cluster) or shut down (left the cluster)
LAST_TRANSITION	The time of the last cluster state transition (when a node joined or left the cluster)
CL_MEMBERS	The number of members currently in the cluster
HW_TYPE	Each node's hardware type

Tailoring the SHOW CLUSTER Display

Adding and Removing Fields

You can add fields to the SHOW CLUSTER display with the ADD command, and you can use the REMOVE command to remove fields you do not need to see.

There are two ways to use the ADD and REMOVE commands:

- If you entered SHOW CLUSTER/CONTINUOUS, type the ADD or REMOVE command, followed by the fields you want to add or remove.
 - You do not need to wait for a prompt; the Command> prompt appears as soon as you start typing.
 - If you want to see the prompt before entering a command, press Ctrl/U.
- Place the commands in a SHOW CLUSTER initialization file. They are then executed automatically whenever you run SHOW CLUSTER.

Example: Adding and Removing Fields

This example shows interactive use of ADD and REMOVE.

```
Command> REMOVE MEMBERS ❶  
Command> ADD TRANSITION_TIME ❷
```

View of Cluster from system ID 62838 node: NEAT 23-SEP-1999
13:21:43

+-----+-----+-----+			
SYSTEMS		MEMBERS	
+-----+-----+-----+			
NODE	SOFTWARE	TRANSITION_TIME	
+-----+-----+-----+			
NEAT	7.2	22-SEP-99 20:38	
TIDY	7.2	22-SEP-99 20:28	
SLOPPY	7.2	21-SEP-99 17:53	
MESSY	HSC V400		
+-----+-----+-----+			

Example Notes: Adding and Removing Fields

Notes on the Adding and Removing Fields example include:

- ❶ This command removes the MEMBERS field from the display.
- ❷ This command adds the TRANSITION_TIME field.

Changing the Width of Fields

Use the SET/WIDTH command to adjust the number of characters that a field occupies in the SHOW CLUSTER display.

Format:

SET field-name /WIDTH=field-width

Example:

SET LAST_TRANSITION /WIDTH=10

SHOW CLUSTER Command Summary

You can control the SHOW CLUSTER display with the following commands:

To do this...	Use this command:
Add a class or field to the display	ADD
Remove a class or field from the display	REMOVE
Change the width or characteristics of a field	SET
Reset the display to the default state	INIT
Enter interactive help mode	HELP
Exit the display	EXIT
Create a command procedure called SHOW_CLUSTER.COM, which recreates the state of your screen	SAVE
Write current data to a file for problem reports	WRITE
Execute a command procedure of SHOW CLUSTER commands	@file-spec

Using a SHOW CLUSTER Initialization File

SHOW CLUSTER Initialization Files

You can customize the SHOW CLUSTER display by creating an initialization file.

The commands in this file are executed whenever SHOW CLUSTER is invoked.

The utility adds or removes the fields specified in the initialization file.

Creating and Using an Initialization File

To create and specify a SHOW CLUSTER initialization file:

1. Create a text file containing SHOW CLUSTER commands (usually ADD, REMOVE, and SET/WIDTH commands).

There are two ways to do this:

- Create the file with any text editor.
 - Enter the SHOW CLUSTER/CONTINUOUS command. Use ADD, REMOVE, and SET/WIDTH commands to tailor the display interactively. When the display is the way you want it, enter the SAVE command. SAVE creates an initialization file that captures the changes you made.
2. Define the logical name SHOW_CLUSTER\$INIT to point to the initialization file. You will probably want to place this definition in your LOGIN.COM procedure.

Example: SHOW CLUSTER Initialization File

Here is an example of a SHOW CLUSTER initialization file:

```
INITIALIZE
REMOVE SYSTEMS
REMOVE SYSTEMS /ID = %X00000000FCD7
REMOVE SYSTEMS /ID = %X000000007311
ADD CL_MEMBERS, LAST_TRANSITION, NODE, HW_TYPE
SET CL_MEMBERS /WIDTH = 6
```

This initialization file produces the following display whenever you enter the SHOW CLUSTER command:

View of Cluster from system ID 69397 node: HANDLE 23-SEP-1999 13:21:43

SYSTEMS		MEMBERS	CLUSTER	
NODE	HW_TYPE	STATUS	CL_MEMB	LAST_TRANSITION
HANDLE	VAXstation 3100/GPX	MEMBER	5	16-SEP-99 09:43
JELLY	VAX 8550	MEMBER		
WIZARD	VAX 6000-420	MEMBER		



Note

Note the /WIDTH qualifier to the SET command for the CL_MEMBERS field of the CLUSTER class.

Monitoring with DCL Commands

Overview

The DCL interface provides commands that are useful for observing the system, such as the `SHOW SYSTEM` and `SHOW PROCESS` commands.

Topics

In this section the following topics are discussed:

- Monitoring memory resources
- Using `SHOW SYSTEM` in a cluster
- Monitoring the activity of a process

Monitoring Memory Resources

Memory Resources

System performance is strongly dependent on the demand for physical memory and the amount of memory installed in the system.

For the OpenVMS system to manage process and system virtual memory, it must divide the available physical memory among different processes and different parts of the operating system.

Page Size in Memory

OpenVMS systems allocate and deallocate memory for processes in units called **pages**. A difference in page size exists between VAX and Alpha systems.

Recognizing this difference is significant to system managers who need to account for additional resources (especially memory resources) that users might require.

- On VAX systems, a page is 512 bytes.
- On Alpha systems, a page size can be one of four values: 8 kilobytes (KB), 16 KB, 32 KB, or 64 KB. A particular Alpha system implements only one of the four page sizes.
- On Alpha systems, a pagelet is a 512-byte unit memory. One Alpha pagelet is the same size as one VAX page. On an Alpha computer with 8 KB pages, 16 Alpha pagelets equal 1 Alpha page.

SHOW MEMORY Command

The SHOW MEMORY command displays information about the physical memory of the system and the disk files used in the management of virtual memory.



Note

When the values of the numbered items in the Free column become too small, system problems occur. To solve these problems may require adding more physical memory, or tuning the system by adjusting various SYSGEN parameters. A temporary measure is to decrease working set sizes for all processes. You might also want to decrease the number of processes allowed in memory at the same time. However, this may increase the number of outswapped processes unless you also reduce the total number of processes allowed on the system at a time.

Example: SHOW MEMORY Command on an OpenVMS VAX System

Here is an example of a SHOW MEMORY display as seen on an OpenVMS VAX system. Try entering this command on your system.

\$ SHOW MEMORY

System Memory Resources on 27-SEP-1999 11:11:08.02

Physical Memory Usage (pages):	Total	Free	In Use	Modified
Main Memory (160.00Mb)	①327680	②251106	③64257	④12317
Virtual I/O Cache Usage (pages):	Total	Free	In Use	Maximum
Cache Memory	4884	124	4760	284010
Slot Usage (slots):	Total	Free	Resident	Swapped
Process Entry Slots	100	54	46	0
Balance Set Slots	90	46	44	0
Dynamic Memory Usage (bytes):	Total	Free	In Use	Largest
Nonpaged Dynamic Memory	3100160	100992	2999168	12544
Paged Dynamic Memory	1379840	913360	466480	910576
Paging File Usage (pages):		Free	Reservable	Total
DISK\$OPENVMS_60:[SYS0.SYSEX]PAGEFILE.SYS		38000	⑤-68291	38000

Of the physical pages in use, 27508 pages are permanently allocated to OpenVMS.

Example Notes: SHOW MEMORY Command on an OpenVMS VAX System

Notes on the SHOW MEMORY Command on an OpenVMS VAX System include:

- ① 327680 pages equate to 160 MB of physical memory.
- ② There are 251106 pages currently free for use.
- ③ There are 64257 pages in use.
- ④ There are 12317 pages that have been modified (written) and are not currently in use.
- ⑤ Swapping file and paging file free as compared to the total is important.

The Reservable field for paging and swapping files is the mechanism for allocating space in these files.

- It represents the difference between the total file size and the number of pages reserved in the file by processes as their virtual address spaces expand.
- Do not be alarmed if it is negative, because only a fraction of the reserved pages are used at one time. (The situation is analogous to an overbooked airline flight.)
- Be most concerned about the Free portion. If it drops to a small fraction of the total file size, run AUTOGEN to increase the size of the file or install another paging file.

Example: SHOW MEMORY Command on an OpenVMS Alpha System

Here is an example of a SHOW MEMORY display as seen on an OpenVMS Alpha system. Remember that page size is shown in CPU-specific units (8192 bytes on an Alpha computer with 8 KB pages).

\$ SHOW MEMORY

System Memory Resources on 22-SEP-1999 15:17:29.09

Physical Memory Usage (pages):	Total	Free	In Use	Modified
Main Memory (96.00Mb)	❶12288	5561	5936	791
Virtual I/O Cache (Kbytes):	Total	Free	In Use	
Cache Memory	3200	0	3200	
Granularity Hint Regions (pages):	Total	Free	In Use	Released
Execlet code region	512	0	357	155
Execlet data region	128	0	72	56
S0/S1 Executive data region	176	0	176	0
S2 Executive data region	50	0	50	0
Resident image code region	1024	185	839	0
Slot Usage (slots):	Total	Free	Resident	Swapped
Process Entry Slots	80	53	27	0
Balance Set Slots	78	53	25	0
Dynamic Memory Usage (bytes):	Total	Free	In Use	Largest
Nonpaged Dynamic Memory	1613824	47488	1566336	10240
Paged Dynamic Memory	1597440	1003008	594432	1001568
Buffer Object Usage (pages):		In Use	Peak	
32-bit System Space Windows (S0/S1)		8	19	
32-bit System Space Windows (S2)		0	0	
Paging File Usage (blocks):		Free	Reservable	Total
DISK\$ALPHAVMSFT3:[SYS0.SYSEXEC]SWAPFILE.SYS		14976	14976	14976
DISK\$ALPHAVMSFT3:[SYS0.SYSEXEC]PAGEFILE.SYS		204672	108576	204672
(Page file name not available)		102272	33696	102272

Of the physical pages in use, 1937 pages are permanently allocated to VMS.

Example Notes: SHOW MEMORY Command on an OpenVMS Alpha System

Notes on the SHOW MEMORY Command on an OpenVMS Alpha System example include:

- ❶ 12288 pages equate to 96 MB of physical memory.

Using SHOW SYSTEM in a Cluster

SHOW SYSTEM Command

The SHOW SYSTEM command displays status information about processes on the system.

Qualifiers Useful in a Cluster

On an OpenVMS Cluster system, there are two qualifiers that enhance the usefulness of SHOW SYSTEM.

To display information about...	Use this qualifier:
All processes on all nodes in an OpenVMS Cluster system	/CLUSTER
All processes on the specified node or nodes	/NODE

Example: SHOW SYSTEM /CLUSTER on an OpenVMS System

This example shows the use of SHOW SYSTEM/CLUSTER to display information from each node in the cluster.

\$ SHOW SYSTEM/CLUSTER

OpenVMS 7.2 on node NEAT 23-SEP-1999 10:43:15.61 Uptime 0 12:40:44

Pid	Process Name	State	Pri	I/O	CPU	Page	flts	Pages
22400041	SWAPPER	HIB	16	0	0 00:00:19.91	0	0	0
22400084	SQLSRV\$SERVER	HIB	8	543	0 00:00:05.85	718	55	
22400046	CONFIGURE	HIB	10	59	0 00:00:32.86	523	155	
22400047	IPCACP	HIB	10	8	0 00:00:00.63	111	68	
22400108	RPC\$SWL	HIB	9	32	0 00:00:00.70	125	192	
22400049	ERRFMT	HIB	7	620	0 00:00:18.54	99	101	
2240004A	CACHE_SERVER	HIBO	16	--	swapped out --		99	
2240004B	CLUSTER_SERVER	HIB	9	10	0 00:00:00.79	121	268	
2240004C	OPCOM	HIB	9	725	0 00:00:17.65	579	138	

.
.
.

OpenVMS 7.2 on node SLOPPY 23-SEP-1999 10:43:16.99 Uptime 22 21:05:05

Pid	Process Name	State	Pri	I/O	CPU	Page	flts	Pages
20E00041	SWAPPER	HIB	16	0	0 00:00:09.35	0	0	0
20E00046	CONFIGURE	HIB	10	109	0 00:00:00.31	333	162	
20E00047	IPCACP	HIB	10	8	0 00:00:00.14	116	69	
20E00049	ERRFMT	HIB	7	20809	0 00:02:53.97	103	84	
20E0004A	CACHE_SERVER	HIB	16	250	0 00:00:00.67	64	99	
20E0004B	CLUSTER_SERVER	CUR	8	538	0 00:00:13.85	188	255	
20E0004C	OPCOM	HIB	8	507	0 00:02:11.78	877	85	
20E0004D	AUDIT_SERVER	HIB	10	592	0 00:00:09.41	2766	179	
20E0004E	JOB_CONTROL	HIB	10	350	0 00:00:01.76	353	162	

.
.
.

OpenVMS 7.2 on node TIDY 23-SEP-1999 10:43:16.48 Uptime 0 14:04:26

Pid	Process Name	State	Pri	I/O	CPU	Page	flts	Pages
22200041	SWAPPER	HIB	16	0	0 00:00:04.57	0	0	0
22200046	CONFIGURE	HIB	10	76	0 00:00:00.22	116	162	
22200047	IPCACP	HIB	10	8	0 00:00:00.10	105	148	
22200049	ERRFMT	HIB	7	490	0 00:00:04.24	85	121	
2220004A	CACHE_SERVER	HIB	16	17	0 00:00:00.12	64	99	
2220004B	CLUSTER_SERVER	CUR	8	17	0 00:00:00.47	133	313	
2220004C	OPCOM	HIB	8	109	0 00:00:04.25	272	144	
2220004D	AUDIT_SERVER	HIB	10	101	0 00:00:01.42	1362	372	
2220004E	JOB_CONTROL	HIB	10	53	0 00:00:00.55	146	228	

.
.
.

\$

Example: SHOW SYSTEM/NODE on an OpenVMS VAX System

This example shows the use of SHOW SYSTEM/NODE on an OpenVMS VAX system to display a SHOW SYSTEM output from each of the specified nodes. Try using this command on your own system, specifying the names of nodes in your cluster.

```
$ SHOW SYSTEM /NODE=(NEAT,TIDY)
```

```
OpenVMS 7.2 on node NEAT 23-SEP-1999 10:43:57.99 Uptime 0 12:41:26
```

Pid	Process Name	State	Pri	I/O	CPU	Page	flts	Pages
22400041	SWAPPER	HIB	16	0	0 00:00:19.91	0	0	0
22400084	SQLSRV\$SERVER	HIB	8	543	0 00:00:05.85	718	55	
22400046	CONFIGURE	HIB	10	59	0 00:00:32.90	523	155	
22400047	IPCACP	HIB	10	8	0 00:00:00.63	111	68	
22400108	RPC\$SWL	HIB	9	32	0 00:00:00.70	125	192	
22400049	ERRFMT	HIB	7	620	0 00:00:18.54	99	101	
2240004A	CACHE_SERVER	HIBO	16	--	swapped out --			99
2240004B	CLUSTER_SERVER	HIB	9	10	0 00:00:00.79	121	268	
2240004C	OPCOM	HIB	7	727	0 00:00:17.71	587	146	

```
.  
.
.
```

```
OpenVMS 7.2 on node TIDY 23-SEP-1999 10:43:59.00 Uptime 22 20:58:23
```

Pid	Process Name	State	Pri	I/O	CPU	Page	flts	Pages
21400041	SWAPPER	HIB	16	0	0 00:00:16.31	0	0	
21400046	CONFIGURE	HIB	10	113	0 00:00:00.50	174	160	
21400049	ERRFMT	HIB	7	16773	0 00:03:27.52	85	121	
2140004A	CACHE_SERVER	HIB	16	123	0 00:00:00.42	64	99	
2140004B	CLUSTER_SERVER	CUR	8	204	0 00:00:12.55	141	332	
2140004C	OPCOM	HIB	8	770	0 00:02:13.52	813	146	
2140004D	AUDIT_SERVER	HIB	10	689	0 00:00:08.23	1668	203	
2140004E	JOB_CONTROL	HIB	10	532	0 00:00:02.12	229	167	

```
.  
.
.
```

```
$
```

Monitoring the Activity of a Process

Monitoring Process Activity

Use the `SHOW PROCESS/CONTINUOUS` command to determine whether a particular process is running and to display its characteristics. Much of the information is the same as that shown by the `SHOW SYSTEM` command, except that the information is for only *one* process and is continuously updated.



Note

The new `/THREADS` qualifier displays the hardware and software context of all the threads associated with the current process.

SHOW PROCESS /CONTINUOUS Command

To monitor any process on the system, enter the `SHOW PROCESS` command.

Format:

`SHOW PROCESS/CONTINUOUS/IDENTIFICATION=pid`

Example:

```
$ SHOW PROCESS/CONTINUOUS/IDENTIFICATION=2020AB4F
```

If the process runs under a UIC that is in your UIC group, you can specify it by name.

Format:

`SHOW PROCESS/CONTINUOUS process-name`

If there are spaces or lowercase letters in the process name, enclose it in quotes.

```
$ SHOW PROCESS/CONTINUOUS "The Wizard"
```

To stop monitoring the process, press `Ctrl/Z` or type the letter `E`.

Privileges Required

The use of the SHOW PROCESS/CONTINUOUS command may require privilege.

To display a process under...	You need this privilege:
Your own UIC	No privilege
Another UIC in your UIC group	GROUP or WORLD
Any UIC outside your UIC group	WORLD



Note

You can omit leading zeros when specifying a PID.

The /MEMORY qualifier is only allowed on the current process.

Example: SHOW PROCESS /CONTINUOUS

Here is an example of the output from SHOW PROCESS/CONTINUOUS. Because the display is frequently updated, try this command on your own system and observe the output.

```
$ SHOW PROCESS/CONTINUOUS/IDENTIFICATION=000007CC
```

```

                                Process USER3      ❶                               17:27:29

State                          LEF  ❷                               Working set           861
Cur/base priority             9/4  ❸                               Virtual pages         3899
Current PC                     7FFEE44C                               CPU time              00:10:43.48  ❹
Current PSL                    03C00004                               Direct I/O            3691
Current user SP                7FEF9EBC                               Buffered I/O          26600
PID                            000007CC                               Page faults           58835
UIC                            [GROUP11,USER3]                               Event flags           E0000043
                                                                D4000002

```

```
$1$DUA0: [SYS1.SYSCOMMON.] [SYSEXE]MAIL.EXE  ❺
```

Example Notes: SHOW PROCESS /CONTINUOUS

Notes on the SHOW PROCESS /CONTINUOUS example include:

- ❶ Name of process
- ❷ Current state
- ❸ Current and base priority
- ❹ CPU time
- ❺ Name of image being run

Using the Monitor Utility

Overview

The Monitor utility displays many different types of data that reflect system performance. Some types are easily interpreted, while some require a knowledge of the internals of the OpenVMS operating system.

You should get into the habit of running MONITOR regularly while the system is running well. This provides a **baseline** of data. Later, if the system encounters a performance problem, you can compare current MONITOR data with the baseline data. Even without a detailed understanding of what the data represent, you can:

- Observe the system to get accustomed to what the displays normally look like.
- Be able to spot an abnormal situation and report it to someone who can do troubleshooting.
- Generate MONITOR reports for someone who can interpret them.

Topics

In this section the following topics are discussed:

- Monitor utility
- Monitoring processes
- Using MONITOR in an OpenVMS Cluster system
- MONITOR command procedures

Monitor Utility

The Monitor utility displays information about system resource usage. Unlike most utilities that display system information, MONITOR can:

- Display several classes of information alternately
- Summarize statistics over a long period of time
- Record information in a disk file
- Play back information that it has recorded

Running the Monitor Utility

Use the MONITOR command to run the Monitor utility.

Format:

MONITOR class-name(s) [/qualifiers]

Example:

```
$ MONITOR
MONITOR>
```

Monitor Commands

At the MONITOR> prompt, you can enter commands that:

- Display a class of information
- Set default classes
- List defaults
- Execute command procedures
- Obtain help
- Exit from the utility

Interrupting the MONITOR Display

While the Monitor utility is displaying information, you can request another display or exit from the utility.

To take this action...	Do this:
Return to the MONITOR> prompt.	Press Ctrl/C.
Exit the utility.	Press Ctrl/Z while a display is running. (or) Enter the EXIT command from the MONITOR> prompt.

Example: MONITOR SYSTEM Display

This class includes several of the most important types of information that you need to monitor. The SYSTEM class presents a general overview of system activity.

If you suspect a performance problem, this is typically the display you look at first.

\$ MONITOR SYSTEM

Node: TIDY
13:21:37

OpenVMS Monitor Utility

22-SEP-1999

Statistic: CURRENT

SYSTEM STATISTICS

				Process States	
		+ CPU Busy (96)	-+	LEF:	5 LEFO: 0
		*****		HIB :	19 HIBO: 0
CPU	0	+-----+ 100		COM:	2 COMO: 0
		*****		PFW:	0 Other: 1
		+-----+		MWAIT:	0
		Cur Top: BATCH_1036 (56)		Total:	27
		+ Page Fault Rate (108)	-+	+ Free List Size (76414)	-+
		*****		*****	89K
MEMORY	0	+-----+ 100		0 +-----+	
		*****		*****	2621
		+-----+		+ Modified List Size (917) +	
		Cur Top: MARCH (106)			
		+ Direct I/O Rate (52)	-+	+ Buffered I/O Rate (4)	-+

I/O	0	+-----+ 60		0 +-----+ 150	

		+-----+		+-----+	
		Cur Top: BATCH_1036 (47)		Cur Top: MARCH (4)	

Example Notes: MONITOR SYSTEM Display

Notes on the MONITOR SYSTEM Display example include:

- The display is updated regularly (every 3 seconds by default).
- Each value is averaged over the sampling interval (3 seconds by default) to yield a rate.
- For certain displays (SYSTEM, CLUSTER, ALL_CLASSES) the default is 6 seconds.



Note

Once you know which resource (memory, I/O, or CPU) is limited, you can use other MONITOR displays to pinpoint the problem.

Adjusting the Sampling Interval

The Monitor utility samples data every 3 seconds by default (6 seconds for the SYSTEM and ALL_CLASSES classes).

To adjust the rate at which it samples data, use the /INTERVAL qualifier, specifying the number of seconds in each sampling interval.

Example:

```
$ MONITOR SYSTEM /INTERVAL=3
```

Adjusting the Display Interval

The rate at which the Monitor utility updates the display defaults to the sampling interval. You may wish to adjust the display interval; for example, you may want to lengthen it if your terminal is connected through a low-speed line.

To adjust the display interval, use the /VIEWING_TIME qualifier, specifying the number of seconds between updates to the display.

Example:

```
$ MONITOR SYSTEM /VIEWING_TIME=10
```

Creating a Recording File

The /RECORD qualifier creates a binary **recording file** that allows future analysis using the same data with various classes and qualifiers.

- All data for the classes you specify is recorded.
- Data is written once per sampling interval.
- If you do not want to view the display while recording is taking place, use /NODISPLAY in conjunction with /RECORD.

```
$ MONITOR/RECORD=MONDAY_MONITOR.DAT/NODISPLAY  
DISK, PROCESSES
```

Creating a Summary File

The /SUMMARY qualifier creates a text file, called a **summary file**, containing summary statistics on all data collected for one MONITOR request.

The summary file contains one page of output for each requested class. The format of each page of a regular summary report is similar to that of display output.

Example:

```
$ MONITOR/SUMMARY=MONDAY_SUMMARY.TXT DISK, PROCESSES
```

Monitoring Processes

PROCESSES Class

The PROCESSES class displays information about system resource usage by individual processes. It identifies the processes that are the heaviest consumers of those resources.

Specifying Resources to Monitor

There are four qualifiers used most often with the PROCESSES class to specify the resources to be monitored.

To display this information...	Use this qualifier:
Top buffered I/O users	/TOPBIO
Top direct I/O users	/TOPDIO
Top CPU users	/TOPCPU
Top page fault users	/TOPFAULT



Note

Warn students against trying to interpret these displays if they do not have enough background to do so. Avoid discussing system performance concepts in detail, because they are covered in a later course.

Only discuss the following two items if you feel comfortable with the material.

Beginning with OpenVMS V6, a new qualifier, /TOPRBS, has been added to the MONITOR PROCESSES command. Using the qualifier /TOPRBS produces a bar graph listing the top balance slot faulting processes.

Beginning with OpenVMS V6, the MONITOR VBS command displays information about the processing of virtual balance slots (VBS).

Example: MONITOR PROCESSES /TOPCPU

Here is an example of the PROCESSES/TOPCPU class display. It displays the approximate percentage of CPU time being used by up to eight of the heaviest CPU users.

```
$ MONITOR PROCESSES /TOPCPU
```

```

                                OpenVMS Monitor Utility
                                TOP CPU TIME PROCESSES
                                on node TIDY
                                22-SEP-1999 13:54:12

                                0      25      50      75      100
                                + - - - - + - - - - + - - - - + - - - - +
21200524 BATCH_1036          78 *****
                                |
21200646 USER1              9 *****
                                |
2120083F SYSTEM             3 **
                                |
                                |
                                |
                                |
                                |
                                |
                                + - - - - + - - - - + - - - - + - - - - +

```

Example Notes: MONITOR PROCESSES /TOPCPU

Be careful how you interpret the output. In this example, the batch job could be taking most of the CPU for one of two reasons:

- It has a low priority and the higher-priority processes do not need much CPU time (the normal situation)
- It has a high priority and is blocking other processes from executing (an abnormal situation)

Finding the reason would take further investigation.

Using MONITOR in an OpenVMS Cluster System

CLUSTER Class

The CLUSTER class includes information about the three most important resources (CPU, memory, and I/O) and their usage on all nodes of the cluster.

You can use this display to determine at a glance whether one node of the cluster is more heavily loaded than the others.

Monitoring the CLUSTER class requires the DECnet software to be running.



Note

The purpose of this material is to introduce the concepts of monitoring. The following restriction may be beyond the scope of this course and is provided only for your information.

Be aware that remote monitoring between nodes in a mixed-version OpenVMS Cluster is supported only between certain versions of Alpha and VAX systems. If a node cannot be monitored remotely because of version incompatibility, an error message is displayed.

Example: MONITOR CLUSTER

Here is an example of the CLUSTER class display.

```
$ MONITOR
```

```
MONITOR> MONITOR CLUSTER
```

```
%MONITOR-I-ESTABCON, establishing connection to remote node(s)...
```

```
Statistic: CURRENT
```

OpenVMS Monitor Utility
CLUSTER STATISTICS

```
22-SEP-1999 16:55:47
```

CPU					MEMORY				
CPU Busy	0	25	50	75 100	%Memory In Use	0	25	50	75 100
	+	+	+	+	+	+	+	+	+
HORSE	100	*****			HORSE	94	*****		
TIGER	19	***			TIGER	92	*****		
BARNUM	11	**			LION	64	*****		
BAILEY	5	*			BAILEY	43	*****		
RNGLNG	4				RNGLNG	32	*****		
LION	4				BARNUM	24	****		
BEAR	4				BEAR	22	****		
-----					-----				
DISK					LOCK				
I/O Operation Rate	0	25	50	75 100	Tot ENQ/DEQ Rate	0	125	250	375 500
	+	+	+	+		+	+	+	+
\$1\$DUA0:	6	*			BARNUM	8			
\$1\$DUA0: R	6	*			RNGLNG				
\$1\$DUA1:	2				TIGER				
\$1\$DUA1: R	1				HORSE				
\$1\$DUA2:					LION				
\$1\$DUA3:					BAILEY				



Note

This display requires DECnet software (VPM object) to be running.

Other Qualifiers Useful in a Cluster

Two Monitor utility qualifiers are especially useful in an OpenVMS Cluster system.

To display this information...	Use this qualifier:
Statistics from particular nodes	/NODE
Multinode summary	/BY_NODE

Example: MONITOR /NODE

Here is an example of the output from the /NODE qualifier. In this case, it is used to display disk I/O statistics from two nodes.

(Use of this qualifier requires the DECnet software to be running.)

```
$ MONITOR
MONITOR> MONITOR/NODE=(BARNUM,BAILEY) DISK
%MONITOR-I-ESTABCON, establishing connection to remote node(s)...
```

OpenVMS Monitor Utility
DISK I/O STATISTICS
on node BARNUM
22-SEP-1999 11:25:11

I/O Operation Rate			CUR	AVE	MIN	MAX
\$1\$DUA0:	(CLOWN)	BARNUM_SYS	0.59	6.90	0.00	16.75
\$1\$DUA1:	(CLOWN)	TIGHTROPE	0.00	0.15	0.00	0.94
\$1\$DUA2:	(BARNUM)	FLYING	0.00	0.00	0.00	0.00
\$1\$DUA3:	(BAILEY)	TRAPEZE	2.09	3.31	0.00	11.59

OpenVMS Monitor Utility
DISK I/O STATISTICS
on node BAILEY
22-SEP-1999 11:25:18

I/O Operation Rate			CUR	AVE	MIN	MAX
\$1\$DUA0:	(CLOWN)	BARNUM_SYS	0.00	0.16	0.00	1.89
\$1\$DUA1:	(CLOWN)	TIGHTROPE	0.00	0.05	0.00	0.63
\$1\$DUA2:	(BARNUM)	FLYING	0.00	0.00	0.00	0.00
\$1\$DUA3:	(BAILEY)	TRAPEZE	0.00	0.00	0.00	0.00



Note

- /VIEWING_TIME can be increased for displays of remote systems or when the terminal baud rate is slow (9600 baud requires 6 seconds).
- /INTERVAL, /VIEWING_TIME, /RECORD times may have to be increased for certain displays and situations (customization).
- The defaults (3 seconds and 6 seconds) work for most displays.

Multinode Summary

You can generate a multinode summary for any MONITOR class. This is especially useful for monitoring disk activity, because the total I/O load on a disk is the sum of the loads generated by each node that uses it.

For a side-by-side display of MONITOR data from multiple nodes:

- Use MONITOR to collect data in a file on each node.
- Use MONITOR/SUMMARY/BY_NODE to combine the data files and produce a multinode summary.

Procedure to Record Data for a Multinode Summary

This is a command procedure called RECORD.COM that you can run simultaneously on each node of a cluster. It records 1 hour's worth of data in the DISK class.

```
$! Get the node name to use in the recording file name
$!
$ NODE = F$GETSYI("NODENAME")
$!
$! Record disk I/O rates for the next hour
$!
$ MONITOR /NODISPLAY /END="+1:00" /RECORD='NODE'.MON DISK
```

Procedure to Start Recording on Two Nodes

This is a command procedure called SUBMIT.COM. It submits RECORD.COM on two nodes of a cluster. If there are more than two nodes, the procedure would contain more SUBMIT commands.

```
$! Record data in batch on node BARNUM
$!
$ SUBMIT /QUEUE=BARNUM_BATCH RECORD.COM
$!
$! Record data in batch on node BAILEY
$!
$ SUBMIT /QUEUE=BAILEY_BATCH RECORD.COM
```

Example: Creating a Multinode Summary

After RECORD.COM completes running on each node, you can create a summary listing as follows:

```
$ MONITOR /SUMMARY /BY_NODE /INPUT=(BARNUM.MON, BAILEY.MON) /NODISPLAY DISK
$ TYPE MONITOR.SUM
```

```
----- OpenVMS Monitor Utility
| AVE |   DISK I/O STATISTICS
----- MULTI-FILE SUMMARY
I/O Operation Rate
```

Node:	BARNUM		BAILEY		Row	Row	Row	Row
From:	22-SEP-1999	15:06	22-SEP-1999	15:06	Sum	Average	Minimum	Maximum
To:	22-SEP-1999	16:06	22-SEP-1999	16:06				
\$1\$DUA0:	0.34		0.00		0.3	0.1	0.00	0.34
\$1\$DUA1:	0.02		1.01		1.0	0.5	0.02	1.01
\$1\$DUA2:	1.13		3.48		4.6	2.3	1.13	3.48
\$1\$DUA3:	8.24		12.42		20.6	10.3	8.24	12.42
\$2\$DUA0:	0.00		0.00		0.0	0.0	0.00	0.00
\$2\$DUA1:	0.00		0.00		0.0	0.0	0.00	0.00



Note

This example does not show the entire display because it would require 132 columns.

MONITOR Command Procedures

Command Procedures

Three command procedures provided with the OpenVMS operating system automate the use of the Monitor utility. These are:

- SYS\$EXAMPLES:SUBMON.COM
- SYS\$EXAMPLES:MONITOR.COM
- SYS\$EXAMPLES:MONSUM.COM (VAX systems)

These three command procedures perform more functions than RECORD.COM and SUBMIT.COM.

SUBMON.COM

The command procedure SYS\$EXAMPLES:SUBMON.COM

- Executes the data collection command procedure (MONITOR.COM)
- Is invoked automatically at system startup time when added to SYSTARTUP_VMS.COM

Example: SUBMON.COM

The command procedure MONITOR.COM creates a detached process that runs SUBMON.COM.

```
$ !
$ ! SUBMON.COM (Submit MONITOR.COM file)
$ !
$ ! This command file is to be placed in a cluster-accessible
$ ! directory called SYS$MONITOR. At system startup time, for
$ ! each node, it is executed by SYSTARTUP.COM, following logical
$ ! name definitions for the cluster-accessible directories,
$ ! SYS$MONITOR and MON$ARCHIVE.
$ !
$ !
$ ! Submit detached MONITOR process to do continuous recording.
$ !
$ !
$ RUN SYS$SYSTEM:LOGINOUT.EXE -
    /UIC=[1,4] -
    /INPUT=SYS$MONITOR:MONITOR.COM -
    /OUTPUT=SYS$MONITOR:MONITOR.LOG -
    /ERROR=SYS$MONITOR:MONITOR.LOG -
    /PROCESS_NAME="Monitor" -
    /WORKING_SET=512 -
    /MAXIMUM_WORKING_SET=512 -
    /EXTENT=512 -
    /NOSWAPPING
$ !
$ !
$ ! End of SUBMON.COM
$ !
```

MONITOR.COM

The command procedure SYS\$EXAMPLES:MONITOR.COM

- Collects data at startup time
- Mails a summary report for each reboot
- Archives the data collection file

Example: MONITOR.COM

In the following example, change cluster_manager to your user name or SYSTEM.

```
$ !
$ ! MONITOR.COM (Generate MONITOR recording file)
$ !
$ ! This command file is to be placed in a cluster-accessible
$ ! directory called SYS$MONITOR. At system startup time, for each
$ ! node, it creates in SYS$MONITOR a MONITOR recording file, which
$ ! is updated throughout the life of the boot. It also creates in
$ ! MON$ARCHIVE, a summary file from the recording file of the
$ ! previous boot, along with a copy of that recording file.
$ ! Logical name definitions for both cluster-accessible directories,
$ ! SYS$MONITOR and MON$ARCHIVE, must be included in SYSTARTUP.COM.
$ !
$ SET DEF SYS$MONITOR
$ SET NOON
$ PURGE MONITOR.LOG/KEEP:2
$ !
$ !
$ ! Compute executing node name and recording and summary
$ ! file names (incorporating node name and date).
$ !
$ NODE = F$GETSYI("NODENAME")
$ DAY = F$EXTRACT(0,2,F$TIME())
$ IF F$EXTRACT(0,1,DAY) .EQS. " " THEN DAY = F$EXTRACT(1,1,DAY)
$ MONTH = F$EXTRACT(3,3,F$TIME())
$ ARCHFILNAM = "MON$ARCHIVE:"+NODE+"_MON"+DAY+MONTH
$ RECFIL = NODE+"_MON.DAT"
$ SUMFIL = ARCHFILNAM+".SUM"
$ !
$ !
$ !
$ !
$ ! Check for existence of recording file from previous boot
$ ! and skip summary if not present.
$ !
$ OPEN/READ/ERROR=NORECFIL RECORDING 'RECFIL'
$ CLOSE RECORDING
$ !
$ !
$ ! Generate summary file from previous boot
$ !
$ MONITOR /INPUT='RECFIL' /NODISPLAY /SUMMARY='SUMFIL' -
```

```
ALL_CLASSES, DISK/ITEM=ALL, SCS/ITEM=ALL
$ !
$ !
$ ! Compute subject string and mail summary file to cluster manager
$ !
$ A=""
$ B=" MONITOR Summary "
$ SUB = A+NODE+B+F$TIME()+A
$ MAIL/SUBJECT='SUB' 'SUMFIL' cluster_manager
$ !
$ !
$ ! Archive recording file and delete it from SYSS$MONITOR.
$ !
$ COPY 'RECFIL' 'ARCHFILNAM'.DAT
$ DELETE 'RECFIL';*
$ !
$ NORECFIL:
$ SET PROCESS/PRIORITY=15
$ !
$ !
$ ! Begin recording for this boot. The specified /INTERVAL value
$ ! is adequate for long-term summaries; you may require a smaller
$ ! value to get reasonable "semi-live" playback summaries (at the
$ ! expense of more disk space for the recording file).
$ !
$ MONITOR /INTERVAL=600 /NODISPLAY /RECORD='RECFIL' ALL_CLASSES
$ !
$ !
$ ! End of MONITOR.COM
$ !
```

MONSUM.COM

The command procedure SYS\$EXAMPLES:MONSUM.COM for OpenVMS VAX systems:

- Mails a summary report for a 24-hour period
- Mails another summary report for prime time usage
- Resubmits itself (once invoked manually)

In the following example, change cluster_manager to your user name or SYSTEM.

```
$ !
$ ! MONSUM.COM (Generate cluster multi-file summaries)
$ !
$ ! This command file is to be placed in a cluster-accessible directory
$ ! called SYS$MONITOR and executed at the convenience of the cluster
$ ! manager. The file generates both 24-hour and "prime time" cluster
$ ! multi-file summaries and resubmits itself to run each day at midnight.
$ !
$ SET DEF SYS$MONITOR
$ SET NOON
$ !
$ ! Compute file specification for MONSUM.COM and resubmit the file.
$ !
$ FILE = F$ENVIRONMENT("PROCEDURE")
$ FILE =
F$PARSE(FILE,,, "DEVICE")+F$PARSE(FILE,,, "DIRECTORY")+F$PARSE(FILE,,, "NAME")
$ SUBMIT 'FILE' /AFTER=TOMORROW /NOPRINT
$ !
$ ! Generate 24-hour cluster summary.
$ !
$ !
$ MONITOR/INPUT=(SYS$MONITOR:*MON*.DAT;* ,MON$ARCHIVE:*MON*.DAT;*) -
/NODISPLAY/SUMMARY=MONSUM.SUM -
ALL_CLASSES,DISK/ITEM=ALL,SCS/ITEM=ALL -
/BEGIN="YESTERDAY+0:0:0.00" /END="TODAY+0:0:0.00" /BY_NODE
$ !
$ !
$ ! Mail 24-hour summary file to cluster manager and delete the file from
$ ! SYS$MONITOR.
$ !
$ !
$ !
$ MAIL/SUBJECT="Daily Monitor Cluster-wide Summary" MONSUM.SUM
cluster_manager
$ DELETE MONSUM.SUM;*
$ !
$ ! Generate prime-time cluster summary.
$ !
$ !
$ MONITOR/INPUT=(SYS$MONITOR:*MON*.DAT;* ,MON$ARCHIVE:*MON*.DAT;*) -
/NODISPLAY/SUMMARY=MONSUM.SUM -
ALL_CLASSES,DISK/ITEM=ALL,SCS/ITEM=ALL -
```



```
/BEGIN="YESTERDAY+9:0:0.00" /END="YESTERDAY+18:0:0.00" /BY_NODE
$ !
$ !
$ ! Mail prime-time summary file to cluster manager and delete the file
$ ! from SYSS$MONITOR.
$ !
$ !
$ MAIL/SUBJECT="Prime-Time Monitor Cluster-wide Summary" MONSUM.SUM
cluster_manager
$ DELETE MONSUM.SUM;*
$ !
$ ! End of MONSUM.COM
$ !
```

Monitoring the Network with NCP Commands

Overview

The network control program (NCP) has commands that allow you to monitor the status of certain network items.

Topics

In this section the following topics are discussed:

- Monitoring DECnet lines
- Monitoring DECnet circuits
- Monitoring DECnet links

Monitoring DECnet Lines

DECnet Lines

Lines are the physical data communication paths between nodes.

Identifying Lines

Every line on a node has a unique identifier. A line identifier has the following format:

■ **dev-c[-u]**

The parts of the line identifier are as follows:

Part	Function
dev	Mnemonic for a communications interface device, for example: ■ BNA: Ethernet device on VAXBI bus ■ QNA: Ethernet device on Q-bus ■ SVA: Ethernet device on MicroVAX 2000 system ■ UNA: Ethernet device on UNIBUS ■ MFA: FDDI device on XMI ■ DMB: DMB32 synchronous device ■ DMC: DMC11 or DMR11 synchronous device ■ CI: CI: computer interconnect
c	Represents a decimal number (0 or a positive integer) designating the device's hardware controller
u	Represents a decimal unit or line number (0 or a positive integer) included if the device is a multiple unit line controller

Displaying Lines

The NCP command `SHOW LINE` displays information about any line you specify. In particular, it tells you whether the line is on (available for communication).

To see a list of available lines, use the `SHOW KNOWN LINES` command, as in this example:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN NCP
NCP>SHOW KNOWN LINES
```

Known Line Volatile Summary as of 20-NOV-1999 10:02:00

Line	State
QNA-0	on

Try this command on your own system to find out what lines are normally in operation. Later, if you have a problem with DECnet communication, you can use this command to make sure the lines are still operating.

You can also use `LIST KNOWN LINES` to find out what lines are defined in the permanent database.



Note

Use the `SHOW LINE` command to display line information (from the volatile database) available to the executor node.

Use the `LIST LINE` command to display line information (from the permanent database) available to the executor node.

Monitoring DECnet Circuits

DECnet Circuits

A circuit is a virtual connection, set up by the DECnet software, between a pair of nodes. For certain types of lines, more than one circuit can use the line.

You can use NCP commands to manipulate all circuits connected to the local node or to add a circuit without reconfiguring the node.

Identifying Circuits

Each circuit on a node must have a unique identifier. A circuit identifier has one of the following formats:

- **dev-c**
- **dev-c-u**
- **dev-c.t**

Circuit Identifier

The parts of the circuit identifier are as follows:

Part	Function
dev	Mnemonic for a communications interface device
c	Represents a decimal number (0 or a positive integer) designating the hardware controller for the device
u	Represents a decimal unit or circuit number included only if there is more than one unit associated with the controller
t	Represents a decimal number identifying a tributary on a multipoint circuit



Note

The t represents a logical tributary number, not to be confused with the address used to poll the tributary

Displaying Circuits

The NCP command `SHOW CIRCUITS` displays information about any circuit you specify. In particular, it tells you whether the circuit is on (available for communication) and which node is at the other end of the circuit.

To see a list of available circuits, use the `SHOW KNOWN CIRCUITS` command.

Example: Displaying Circuits

In the following example, QNA-0 is the only circuit.

```
NCP>SHOW KNOWN CIRCUITS
```

Known Circuit Volatile Summary as of 20-NOV-1999 10:04:34

Circuit	State	Loopback Name	Adjacent Routing Node
QNA-0	on		19.1022 (ZKDRB)

Try this command on your own system to find out what circuits are normally in operation. Later, if you have a problem with DECnet communication, you can use this command to make sure the circuits are still operating.

You can also use `LIST KNOWN CIRCUITS` to find out what circuits are defined in the permanent database.

Circuit States

This table describes the states that can be displayed by the `SHOW CIRCUITS` command.

This state...	Has this function:
OFF	Allows no traffic over a circuit. The circuit is unavailable for network activity.
ON	Allows traffic over the circuit. Allows for complete route-through and downline loading operations. This is the normal operational state.
SERVICE	Reserves the circuit for service functions (upline dumping, downline loading, or line-level loopback testing). Only an Ethernet circuit allows logical link activity or route-through at the same time as service operations.
CLEARED	Applies only to X.25 circuits. The X.25 circuit does not exist in the volatile database when PSI is loaded.

Monitoring DECnet Links

DECnet Links

A link is a logical communication path between two processes. Every time a user initiates some network activity (such as using SET HOST or sending mail to another node) the DECnet software creates a link and maintains it as long as necessary.

Displaying Links

The NCP command SHOW LINKS displays information about any link that you specify, including the name of the process that is using the link.

To see a list of available links, use the SHOW KNOWN LINKS command.

This command is useful when you need to stop the network and would like to find out whether any users are currently using it. If you use the SET EXECUTOR STATE OFF command, the DECnet software deletes all existing links.

Example: Displaying Links

This example shows the output from a SHOW KNOWN LINKS command entered on node PANHED.

```
NCP>SHOW KNOWN LINKS
```

Known Link Volatile Summary as of 20-NOV-1999 10:06:03

	Link	Node	PID	Process	Remote Link	Remote user
❶	8317	2.230 (NEAT)	22400057	REMACP	8690	USER1
	8233	3.189 (NEARBY)	22400057	REMACP	16588	USER1
❷	8318	56.110 (FAROUT)	22400145	SYSTEM	34065	CTERM
	8281	61.151 (OTHER)	22400057	REMACP	8225	USER2

```
NCP>
```

Example Notes: Displaying Links

Notes on the Displaying Links example include:

- ❶ Remote user USER1 on node NEAT has entered a SET HOST command to this node. The REMACP process is responsible for the local end of the link.
- ❷ A local process (SYSTEM) has entered a SET HOST command to node FAROUT.

Try the SHOW KNOWN LINKS command on your own system to find out what links are in operation. If there is no network activity, no links are displayed.



Note

Links with certain nodes are specified with the command component WITH NODE node-name-or-address.

```
NCP>SHOW KNOWN LINKS WITH NODE NEAT
```

Known Link Volatile Summary as of 20-NOV-1999 10:07:29

Link	Node	PID	Process	Remote	Link	Remote user
8323	2.230 (NEAT)	21400202	_FTA138:		8576	CTERM

Summary

Concepts

Using the Show Cluster Utility

- The Show Cluster utility displays a variety of information about an OpenVMS Cluster system.
- You can tailor the SHOW CLUSTER display through the use of interactive commands or an initialization file.

Monitoring with DCL Commands

- The SHOW MEMORY command provides information about the use of physical memory.
- The SHOW SYSTEM command has some qualifiers that are useful in an OpenVMS Cluster system.
- Use the SHOW PROCESS command to monitor the activity of a process.
- The Monitor utility displays many different types of data that reflect system performance.
- You should get into the habit of running MONITOR regularly while the system is running well, to gather a baseline of data. Later, if the system encounters a performance problem, you can compare current MONITOR data with the baseline data.
- The PROCESSES class displays information about system resource usage by individual processes. It identifies the processes that are the heaviest consumers of those resources.
- The CLUSTER class includes information about the three most important resources (CPU, memory, and I/O) and their usage on all nodes of the cluster.
- The following command procedures provided with the OpenVMS operating system automate the use of the Monitor utility:
 - SYS\$EXAMPLES:SUBMON.COM
 - SYS\$EXAMPLES:MONITOR.COM
 - SYS\$EXAMPLES:MONSUM.COM (VAX systems)

Monitoring the Network with NCP Commands

- Lines are the physical data communication paths between nodes.
- A circuit is a virtual connection, set up by the DECnet software, between a pair of nodes.
- A link is a logical communication path between two processes.
- Use the NCP SHOW command to display information about:
 - DECnet lines
 - DECnet circuits
 - DECnet links

Commands

Using the Show Cluster Utility

SHOW CLUSTER

Displays a variety of information about the OpenVMS Cluster system.

Monitoring with DCL Commands

SHOW MEMORY

Displays information about memory usage.

SHOW SYSTEM /CLUSTER

Displays SHOW SYSTEM information for every node in the cluster.

SHOW SYSTEM /NODE

Displays SHOW SYSTEM information for specified nodes.

SHOW PROCESS /CONTINUOUS

Continuously displays information about a process.

MONITOR

Runs the Monitor utility.

MONITOR SYSTEM

Displays several important classes of information, providing a general overview of system activity.

MONITOR PROCESSES

Displays information about all processes on the system.

MONITOR CLUSTER

Displays resource usage information about the entire OpenVMS Cluster system.

Monitoring the Network with NCP Commands

SHOW KNOWN LINES

NCP command to show the DECnet lines and whether each one is in operation.

SHOW KNOWN CIRCUITS

NCP command to show the DECnet circuits and whether each one is in operation.

SHOW KNOWN LINKS

NCP command to show the DECnet links and information about each one.

Understanding Security Concepts

Module 8

About This Module

Introduction

Different installations require different levels of system security. The system manager has control over the level of security implemented. System security issues involve:

- Physical security
- Software security

The OpenVMS operating system allows you to manage software security on many levels by restricting user passwords and assigning user privileges.

Topics

This module presents the basic concepts of maintaining system security. Among the topics covered are:

- General system security
- Using identifiers and access control lists
- Restricting user activities
- Using passwords to maintain system security

Objectives

To describe the tasks and responsibilities in maintaining system security, you should be able to:

- List OpenVMS features that provide general system security.
- Explain the user security profile and its contents.
- Restrict user activities to maintain system security.
- Create and control object security.
- Use passwords to maintain system security.

General System Security

Overview

System managers need to be concerned with both the physical and software security of OpenVMS systems.

Topics

In this section the following topics are discussed:

- Physical security considerations
- Software security considerations
- OpenVMS Cluster security considerations

Physical Security Consideration

Physical Security

Different installations require different levels of physical security. The system manager has control over the level of security implemented.

Physical security issues include:

- Access to console terminal (allows user to change system parameters, therefore should be prevented)
- Availability of dialup lines



Note

You should consider the physical security of the computer first because a user with access to the console terminal can bypass software controls. If dialup lines are available, users can attempt to break in to the system from outside, presenting a security risk.

Media storage is also a security issue. Cabinets with locks are available for storing sensitive media as well as less expensive cabinets.

To allow access to the room but not to the media, you must obtain the more expensive cabinets. If users have no need to gain access to the computer or the media, the room should remain locked to prevent unnecessary traffic (traffic brings dust and dirt into the room) and to reduce the cost of maintaining security.

You can disable conversational boot on workstations so they only boot from the main (secure) cluster.

Levels of Required Security

At one extreme, users can be allowed physical access to the computer as well as all privileges to do whatever they choose. At the other extreme, the computer and peripherals could be in a locked room accessible only to operators, and users could have few or no privileges. Conversational startup on workstations can be disabled so users cannot change system parameters.

Between these two extremes, the manager can use the tools available to grant specific levels of access and privilege to specific users. The manager has control over the level of security implemented.

Examples of Physical Security

Examples of physical security are:

- Keeping the console terminal in a locked or restricted area
- Not posting dialup line phone numbers in public locations
- Keeping all media storage in a locked or restricted area

Software Security

Once the computer itself is physically secure, the manager must maintain some level of software security.

Minimum Software Security

To ensure basic minimum security, the system manager should:

- Keep SYSUAF.DAT, RIGHTSLIST.DAT, NETPROXY.DAT, and NET\$PROXY (VAX systems) up to date.
- Use AUTHORIZE flags and hourly restrictions.
- Insist on nontrivial passwords.
- Not publicize dialup numbers.
- Not permit WORLD access to SYSUAF.DAT or RIGHTSLIST.DAT.
- Encourage use of file protection codes and ACLs.
- Restrict user privileges.
- Create accounting reports to check system usage.
- Label disks and tapes in a systematic manner.

Additional Security

If your system needs to be more secure, choose any or all of the security tools provided by the OpenVMS system, including:

- Generated passwords
- Password history lists
- Erase-on-delete and erase-on-allocate for files
- Login security
- Break-in detection
- Captive and restricted accounts to control user activity
- Security auditing
- Protected subsystems
- Secondary passwords

OpenVMS Cluster Security Considerations

Security Management in an OpenVMS Cluster System

An OpenVMS Cluster system must be treated as a single management domain. It must be managed as a single system, by a single system manager, or a cooperating management team. Even if a cluster has many different working environments, the different systems have access to common, shared resources and must be guided by a single security and management policy.



Note

You should not merge systems that have incompatible security needs. If your systems have different needs, you may want to form a network instead of a cluster, or to place only those systems with similar needs into the cluster. In a DECnet for OpenVMS network, any system can be protected so that users logged in to another system cannot get access to files or execute privileged programs, even if those users are privileged.

Building a Common Environment

In a single management domain, a process, acting on behalf of an authorized individual, requests access to a cluster-visible object, and a coordinating node determines the outcome by comparing its copy of the common authorization database with the security profile for the object being accessed.

To achieve data consistency within the cluster, a site needs to:

- Maintain a common set of data
- Execute changes to system parameters consistently

When changing any LGI system parameters, use the System Management utility (SYSMAN).

Required System Files

The easiest way to ensure a single security domain is to maintain a single copy of each of the files on one or more cluster-mounted disks. As soon as any required file is created on one node, it must be created or commonly referenced on all remaining cluster members. When a cluster is configured with multiple system disks, system logical names can be used to ensure that only a single copy of each file exists.

System Files That Must Be Common in a Cluster

System File	Description
NETOBJECT.DAT	Contains the DECnet object database. Among the information contained in this file is the list of known DECnet server accounts and passwords.
NETPROXY.DAT	Contains the network proxy database. This file is maintained by the Authorize utility (AUTHORIZE).
†NET\$PROXY.DAT	Contains the network proxy database. This file is maintained by the Authorize utility (AUTHORIZE)
QMAN\$MASTER.DAT	Contains the master queue manager database. This file contains the security information for all shared batch and print queues. A single copy of this file must be maintained on a shared disk if two or more nodes intend to participate in a shared queuing system.
RIGHTSLIST.DAT	Contains the rights identifier database. This file is maintained by AUTHORIZE and by various rights identifier system services.
SYSALF.DAT	Contains the system autologin file. This file is Maintained by the System Management utility (SYSMAN).
SYSUAF.DAT	Contains the system user authorization file. This file is maintained by AUTHORIZE and modifiable through the Set User Authorization Information (\$SETUAI) system service.
SYSUAFALT.DAT	Contains the system alternate user authorization file. This file serves as a backup to SYSUAF.DAT and is enabled through the SYSUAFALT system parameter.
VMS\$OBJECTS.DAT	Contains the cluster-visible object database. Among the information contained in this file are the security profiles for all cluster-visible objects.
†VAX specific	



Note

Using shared files is not the only way of achieving a single security domain. Some sites may have requirements for multiple copies of one or more of these system files on different nodes in a cluster. As long as the security information available to each node in the cluster is exactly the same, these sites operate in a single security domain. See the *Security Guide* for information on coordinating security data.

User Security Profile

Overview

User processes and applications have security profiles, which carry a consistent set of elements:

- User identification code (UIC)
- Rights identifiers
- Privileges

Before a process can access a file, device, global section, or any other protected object, the operating system checks to ensure that the requesting process has the authority to access the object in the manner requested. To establish this, the operating system examines the security profile of both the requesting process and the object.

Topics

In this section the following topics are discussed:

- User identification code
- Rights identifiers
- Privileges

User Identification Code

The first element of a user's security profile is the user identification code (UIC). The UIC identifies what group you belong to and what your unique identification is within that group.

[300, 301]

In this UIC, the user belongs to group 300 and has the unique identifier of 301.

UIC Group Identifier

A group usually consists of users who work in the same department or on the same project. Group numbers are assigned by the system manager in the group number part of their UICs.

UIC Protection

When you log in, the system creates a process for you and assigns the UIC defined in your UAF record to your process. Thereafter, you have access to files or to other users based on the value of your UIC.

Assigning UICs

Before assigning UICs to users, analyze the extent to which the user processes on your system need to:

- Share access to files, volumes, and devices on your system
- Communicate interactively with each other
- Affect or control each other

To provide...	Assign users to:
Greater interaction	The same UIC group
Greater protection	Different UIC groups

Suggestions for Dividing Users into Groups

The following table suggests some ways of dividing users into groups based on the user need.

To perform this function...	Do this:
Allow specific users access to other users' files and structures, but deny access to other users on the system.	Divide users into groups along project or departmental lines. Assign a different group UIC number to each group. Leave room between each assigned number for additional users. Users within a group can access each other's files with the access set for GROUP in the protection code of the accessed file. Users outside a group access files belonging to a group member with the access defined for WORLD only.
Protect system files, but allow general access to all other files on the system.	Assign all users a UIC of [295,300] or some other common UIC. ¹ All users own all user files and have OWNER access to all user files. Users cannot access system files. ² Divide users into groups. ¹
Identify groups for accounting and observation purposes, but allow general access to all files, including system files.	Modify the system parameter RMS_FILEPROT to set the default protection assigned to all new files so WORLD has complete access. You can collect accounting information on separate users (according to their UICs), but because all files created allow all users complete access, the files are not really protected. In this case, newly created system files are not protected either.
Identify groups for accounting and observation purposes but allow access to all nonsystem files.	Create a LOGIN.COM file for each user and include the SET PROTECTION/DEFAULT=(W:RWED) command in it. Do not include this command in your LOGIN.COM file. You can collect accounting information on separate users (according to their UICs). Users can access all files on the system except your files and system files.
¹ System files usually have a group number between 1 and 10. Do not assign a group number reserved for system users (typically 1-10). Modify the MAXSYSGROUP parameter to set the upper limit of the group number for system users as needed.	
² Users can still gain access through the system or owner field of the protection code, an access control list (ACL), or a privilege.	

Process Interaction

To perform this function...	Do this:
Increase the ability of processes to interact and share information.	Assign process UICs to the same group.
Decrease the ability of processes to interact and share information.	Assign process UICs to different groups.

Effect of Group UIC Assignment

The following table details the effects of UIC group assignment on process interaction.

To produce this desired interaction...	Do this:
Share files, volumes, and devices within the group, while denying access to processes in other groups.	Set the GROUP protection codes of the files, volumes, and devices properly (G:RWED).
Communicate with group members by means of structures that processes in other groups cannot access.	Set logical names and identifiers such as: ■ Group logical names ■ Group mailbox logical names ■ Group global section names ■ Common event flag clusters
Affect and control other processes in group.	Grant the user GROUP privilege.

Rights Identifiers

Rights Identifiers

The second element of a user's security profile is the set of rights identifiers. The rights identifiers:

- Represent an individual user or a group of users
- Allow temporary identification of groups of users by a common identifier for as long as needed

Types of Identifiers

There are five types of identifiers:

- Environmental
- General
- UIC
- Facility
- System node

Environmental Identifiers

Environmental identifiers are defined by the system at login.

- Alphanumeric strings automatically created by the system
- Examples:
BATCH
NETWORK
INTERACTIVE
LOCAL
DIALUP
REMOTE

General Identifiers

General identifiers are defined by the security administrator.

- Alphanumeric strings of 1 to 31 characters with at least one alphabetic character.
- Valid characters include numbers 0 to 9, characters A to Z and a to z, the dollar sign (\$), and the underscore (_).
- Examples:
SALES
PERSONNEL
DATA_ENTRY

UIC Identifiers

UIC identifiers are based on the user's identification code.

- Alphanumeric UICs, with or without brackets.
- Valid characters are the same as those for a general identifier.
- Examples:
[GROUP1,JONES]
[JONES]
GROUP1
JONES

Facility Identifiers

Facility identifiers are defined by the application.

- Valid characters are the same as those for a general identifier.
- Example: DBM\$MOD_SCHEMA

System Node Identifiers

System node identifiers are defined by the system startup procedures.

Example: Displaying Rights Identifiers

Use the SHOW PROCESS/ALL command to display the identifiers for your current process.

```
$ SHOW PROCESS/ALL
10-NOV-1999 13:43:24.91      User:  SARAH      Process ID:  3C404F7B
                             Node:  MOPPET      Process name: "_RTA11:"
```

```
Terminal:      RTA11:
User Identifier: [SARAH] ❶
Base priority:  4
Default file spec: WORK12:[SARAH]
```

Devices allocated: MOPPET\$RTA11:

Process Quotas:

```
.
.
.
```

Process rights:

```
INTERACTIVE ❷
  REMOTE    ❸
  VMS       ❹
  VMS$COM    resource ❺
```

System rights:

```
SYS$NODE_MOPPET ❻
```

Example Notes: Displaying Rights Identifiers

Notes for the Displaying Rights Identifier example include:

- ❶ UIC identifier, indicating user Sarah
- ❷ Environmental identifier, indicating user Sarah is an interactive user
- ❸ Environmental identifier, indicating user Sarah is logged in remotely
- ❹ General identifier, indicating user Sarah is also a member of the VMS group
- ❺ General identifier, indicating user Sarah holds the VMS\$COM identifier with the resource attribute so she can charge disk space to the identifier
- ❻ Environmental identifier, indicating user Sarah is working from the MOPPET node

Privileges

OpenVMS Privileges

The OpenVMS operating system provides a comprehensive set of privileges that allow precise access and control of various resources and operating system facilities.

Typically, the system manager grants only the minimum number of privileges to a typical system user.

Minimum Security Privileges

A typical system user is commonly called a **nonprivileged user**, which means the user has only minimal privileges (as opposed to no privileges at all). The minimum set of required privileges for nonprivileged users can vary from system to system, but typically consists of the following:

- **TMPMBX** (allows the user to spawn subprocesses)
- **NETMBX** (useful only on systems with DECnet services)

Types of Privileges

Two sets of privileges are specified in each UAF record:

- **Authorized** privileges are enabled only by explicit use of the DCL SET PROCESS/PRIVILEGE command.
- **Default** privileges are automatically enabled once the user has logged in.

The reason for having two separate sets of privileges is to avoid accidental misuse of privileges. Sensitive privileges should never be placed in the default privilege set (/DEFPRIVILEGES), but rather in the second privilege set (/PRIVILEGES).

Assigning Privileges

Use the Authorize MODIFY command to assign privileges to users.

To assign this set of privileges...	Use this qualifier:
Default , the list of privileges that are enabled at login time. The keyword [NO]ALL disables or enables all user privileges.	/DEFPRIVILEGES=([NO]privname[,...])
Authorized , the list of privileges granted (but not enabled) at login time.	/PRIVILEGES=([NO]privname[,...])

Categories of Privileges

Privileges are divided into seven categories according to possible damage that users possessing them could cause the system.

- **None** (No privileges)
- **Normal** (Minimum privileges to use the system)
 - NETMBX — Create network device
 - TMPMBX — Create temporary mailbox
- **Group** (Potential to interfere with members of the same group)
 - GROUP — Affect other processes in same group
 - GRPPRV — Gain access through the system protection field of the group's objects
- **Devour** (Potential to consume noncritical systemwide resources)
 - ACNT — Disable accounting
 - ALLSPOOL — Allocate spooled device
 - BUGCHK — Make bugcheck log entries
 - EXQUOTA — Exceed disk quota
 - GRPNAM — Modify group logical name table
 - PRMCEB — Create/delete permanent common event clusters
 - PRMGBL — Create permanent global sections
 - PRMMBX — Create permanent mailbox
 - SHMEM — Create/delete data structures in shared memory
- **System** (Potential to interfere with system operation)
 - ALTPRI — Set base priority higher than allotment
 - AUDIT — Generate audit records
 - OPER — Perform operator functions
 - PSWAPM — Change process swap mode
 - SECURITY — Perform security functions
 - SYSLCK — Lock systemwide resources
 - WORLD — Affect other processes in the world

- **Objects** (Potential to compromise an object's security)
 - DIAGNOSE — Diagnose devices
 - IMPORT — Mount a nonlabeled tape volume
 - MOUNT — Execute mount volume QIO
 - READALL — Possess read access to all system objects
 - SYSGBL — Create systemwide global sections
 - VOLPRO — Override volume protection
- **All** (Potential to control the system)
 - BYPASS — Disregard protection
 - CMEXEC — Change mode to executive
 - CMKRNL — Change mode to kernel
 - DETACH — Create detached processes
 - DOWNGRADE — Write to a lower secrecy object or lower an object's classification
 - LOG_IO — Perform logical I/O
 - PFNMAP — Map to specific physical pages
 - PHY_IO — Perform physical I/O
 - SETPRV — Enable any privilege
 - SHARE — Access devices allocated to other users
 - SYSNAM — Modify system logical name table
 - SYSPRV --- Access objects through the system protection field
 - UPGRADE — Write to a higher integrity object or raise an object's integrity level



Note

- Run through the list of privileges in the tables taking special note of those privileges that you feel would be of most interest to your particular student group.
 - Emphasize the power of the privileges. They should avoid giving privileges unless absolutely necessary.
 - Even for experienced system managers, privileges tend to be of two kinds: normal and DANGEROUS! This is easy to recognize when allotting privileges to users of the system. It is just as important for harassed system managers, in the middle of three tasks at once, to remember where they are (current default directory) and whether privileges are enabled.
 - Take special care to mention that a user with the SETPRV privilege can enable any privilege.
-

Possible Privilege Problems

The following are examples of common problems users could cause given assigned inappropriate privileges:

- A user with the SETPRV privilege can enable any privilege.
- A user with the EXQUOTA privilege has the potential of consuming an entire disk.
- A user with other Devour privileges has the potential of consuming all of the nonpaged pool.
- A user with ALTPRI could raise his or her priority and block all other users.
- A user with WORLD privilege could stop all other processes.

Restricting User Activities

Overview

You can set up user accounts with only the amount of system access and privileges that the individual user needs. In addition, you can restrict users to the use of certain software.

Topics

In this section the following topics are discussed:

- Tailoring user accounts
- Setting access and security fields
- Setting login flags
- Creating captive accounts



Note

The prerequisite course described the very basic steps needed to add a new system user. The parameters used to create the UAF record were quite simplified. Normally you need to consider and add many more values to a new user's UAF record. This section goes into these values in more detail.

Tailoring User Accounts

There are many qualifiers available for use in creating and modifying UAF records with the Authorize utility. These qualifiers provide specific control in defining the user account and processes.

UAF record fields fall into four basic categories:

- Identification and environment
- Access and security
- Quotas and resource limits
- Privileges

Example: UAF Field Categories

The following example shows a sample listing of a UAF record and indicates the location of these field categories.

UAF> SHOW USER1

❶ Username:	USER1	Owner:	Admin user
Account:	GRP11	UIC:	[11,2]
([USER1])		Tables:	DCLTABLES
CLI:	DCL		
Default:	DISK\$USER:[USER1]		
LGICMD:	SYSS\$MANAGER:GRP11LOGIN		
❷ Login Flags:	Diswelcome Disnewmail		
Primary days:	Mon Tue Wed Thu Fri		
Secondary days:		Sat Sun	
Primary	000000000001111111112222	Secondary	000000000001111111112222
Day Hours	012345678901234567890123	Day Hours	012345678901234567890123
Network:	-----#####-----		----- No access -----
Batch:	-----#####-----		----- No access -----
Local:	##### Full access #####		##### Full access #####
Dialup:	-----#####-----		----- No access -----
Remote:	-----#####-----		----- No access -----
Expiration:	(none)	PWDMINIMUM:	6 Login Fails: 0
Pwdlifetime:	90 00:00	Pwdchange:	(pre-expired)
Last Login:	(none) (interactive),		(none) (non-
interactive)			
❸ Maxjobs:	0 Fillm:	20 Bytln:	4096
Maxacct jobs:	0 Shrfillm:	0 Pbytln:	0
Maxdetach:	0 BIOln:	6 JTquota:	1024
Prclm:	2 DIOln:	6 WSdef:	1024
Prio:	4 ASTln:	10 WSquo:	2048
Queprio:	4 TQElm:	10 WSextent:	4096
CPU:	(none) Enqlm:	10 Pgflquo:	10000
❹ Authorized Privileges:			
	GROUP TMPMBX NETMBX		
Default Privileges:			
	TMPMBX NETMBX		

Example Notes: UAF Field Categories

Notes for the UAF field Categories example include:

- ❶ Identification and environment
- ❷ Access and security
- ❸ Quotas and resource limits
- ❹ Privileges

Access and Security Fields

Use access and security fields to:

- Limit the use of the account by time/day or access mode
- Limit certain capabilities once logged in to the system
- Authenticate user requests for access to files and other resources

Access Times and Modes

You can limit an account's access to the system in three ways:

- Time of day
- Day of week
- Access mode

Restricting Access by Time of Day

The following commands allow USER1 to gain access to the system in any manner from 8 A.M. through 5 P.M. except during lunch on primary days, but prevent any access on secondary days:

```
UAF> MODIFY USER1/ACCESS=(PRIMARY, 8-11, 13-16)
UAF> MODIFY USER1/NOACCESS=SECONDARY
```

Restricting Access by Day of Week

The following command sets the primary days to be Monday through Friday for the user USER1. (The other days of the week automatically become secondary days for this user.)

```
UAF> MODIFY USER1/PRIMEDAYS=(MON,TUE,WED,THU,FRI)
```

Restricting Access by Access Mode

You can combine values to further specify the account's ability to gain access to the system. For example, the following commands allow USER1 to use a local terminal during the day and a dialup terminal during evenings and weekends:

```
UAF> MODIFY USER1/LOCAL=(PRIMARY, 8-17, SECONDARY, 8-17)
UAF> MODIFY USER1/DIALUP=(PRIMARY, 18-7, SECONDARY, 0-23)
```

Login Access Modes

The following table describes the various types of access modes.

This access mode...	Permits this type of access:
INTERACTIVE	Any kind of interactive login
LOCAL	Directly connected terminals
DIALUP	Modem connections using telephone services (or network services emulating telephone services)
REMOTE	Virtual terminal connection across DECnet system
BATCH	Batch jobs (noninteractive access method)
NETWORK	DECnet noninteractive network access, for example: file transfers, electronic mail to/from other nodes, and so forth

AUTHORIZE Qualifiers for Access Fields

The following table summarizes the AUTHORIZE qualifiers used to restrict access.

Use this qualifier...	To specify:
/ACCESS[=(range[,...])]	Hours of access for all modes of access.
/BATCH[=(range[,...])]	Hours of access permitted for batch jobs.
/DIALUP[=(range[,...])] ¹	Hours of access permitted for dialup jobs.
/INTERACTIVE[=(range[,...])]	Hours of access permitted for interactive logins.
/LOCAL[=(range[,...])] ¹	Hours of access permitted for interactive logins initiated on local terminals.
/PRIMEDAYS=[([NO]day[,...])]	Primary and secondary days of the week for logins. Specify primary days as MON, TUE, WED, THU, FRI, SAT, and SUN. Specify secondary days as NOMON, NOTUE, NOWED, and so on.
/NETWORK[=(range[,...])]	Hours of access permitted for network batch jobs.
/REMOTE[=(range[,...])] ¹	Hours of access permitted for interactive logins initiated by network remote terminals.
¹ These are interactive logins, so use the /INTERACTIVE qualifier to specify all three interactive access methods.	

Security Fields

Use security fields to authenticate user requests for access to files and other resources.

The following table lists the qualifiers used to set these fields and control the rights database.

Use this qualifier...	To perform this function:
/ADD_IDENTIFIER	Add identifiers for the user name and account name to the rights database.
/EXPIRATION=time	Specify expiration date and time of the account.
/GENERATE_PASSWORD[=keyword]	Invoke the password generator to generate user passwords. Details of the possible keywords are discussed in the OpenVMS Guide to System Security
/MODIFY_IDENTIFIERS	Specify whether the identifier associated with a user record is to be modified in the rights database.
/PASSWORD=(pwd1[,pwd2])	Specify the primary and optional secondary passwords.
/PWDEXPIRED	Specify whether a password is valid only for the first login.
/PWLIFETIME=time	Specify the length of time a password is valid, entered as a delta-time value.
/PWDMINIMUM=value	Specify the minimum number of characters allowed for a password.
/REMOVE_IDENTIFIER	Specify whether the user name and account name identifiers should be removed from the rights database when the UAF record is removed from SYSUAF.DAT (works only for the REMOVE command).

Login Flags

Use login flags to restrict certain activities of the user's job.

Login Flag Parameters

The following table lists the login flag parameters that are related to security; it is *not* a complete list of the login flag parameters.

To do this...	Use this /FLAG parameter: ¹
Audit all security-relevant actions.	AUDIT
Restrict an account to autologins only.	AUTOLOGIN
Prevent user from changing any defaults at login.	CAPTIVE
Prevent user from changing default CLI or CLI table.	DEFCLI
Disable Ctrl/Y interrupt until SET CONTROL_Y is executed.	DISCTLY
Disable forced user-expired password changes at login.	DISFORCE_PWD_CHANGE
Prevent Mail delivery to this user.	DISMAIL
Suppress New Mail... announcement.	DISNEWMAIL
Disable automatic screening of new passwords against a system dictionary.	DISPWDDIC
Disable automatic checking of new passwords against a list of the user's passwords from the last year.	DISPWDHIS
Disable automated reconnections.	DISRECONNECT
Disable time of last login and other security reports.	DISREPORT
Disable this account completely (if your default account has the DISUSER flag set, you must add the /FLAG=NODISUSER qualifier when creating a new account).	DISUSER
Suppress Welcome to... login message.	DISWELCOME
Require user to use generated passwords.	GENPWD
Prevent user from changing password.	LOCKPWD
Mark password as expired.	PWD_EXPIRED
Mark second password as expired.	PWD2_EXPIRED
Ensure that all commands are executed in the system and user login command procedures, and any procedures invoked by these two procedures.	RESTRICTED

¹Any flag can be prefixed with NO to turn off the flag's intended purpose, for example: /NOLOCKPWD.

Creating and Using Captive Accounts

Captive Accounts

Some users may only need access to certain limited software, or they may perform work that is limited to a particular function. They do not require full access to the OpenVMS system. You can give such a user a **captive** account. This user's LOGIN.COM accesses the appropriate software, and exiting from the software causes the user to be logged out.

Characteristics of Captive Accounts

A captive account:

- Prevents the user from changing any defaults at login, for example; /CLI, /DISK, /COMMAND, or /LGICMD.
- Prevents the user from escaping the captive login command procedure and gaining access to the DCL command level.
- Establishes an environment where Ctrl/Y interrupts are initially turned off. Command procedures can still turn on Ctrl/Y interrupts with the DCL command SET CONTROL=Y.

Steps for Setting Up a Captive Account

To set up a captive account:

Step	Task
1	Create a file in the SYS\$MANAGER directory, such as SYS\$MANAGER:CAPTIVE.COM, to customize the user environment and prevent a user from changing it. The captive command procedure must contain a loop to prevent it from exiting.
2	• Run the Authorize utility and: • Set the captive user's LGICMD to contain the name of the captive command procedure or a logical name translating to the captive command procedure name. • Set the FLAGS field of the UAF record to specify the CAPTIVE flag. By default, an account is not captive (NOCAPTIVE). • Assign a unique UIC group to ensure that the captive account users can access only world-accessible files and files owned by the captive account. • Either require no password (/NOPASSWORD) or lock the password (/FLAGS=LOCKPWD) so that only the security administrator can change it • Set the subprocess limit PRCLM to 0, thus preventing the user from spawning out of the account. Be sure to verify that the system parameter PQL_MPRCLM (the minimum subprocess limit) is set to 0.

Example: Setting Up a Captive Account

In the following example, the SYS\$MANAGER:CAPTIVE.COM command procedure calls an application program for the SPECULATE user.

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> MODIFY SPECULATE/FLAGS=(CAPTIVE, LOCKPWD)/PRCLM-
_UAF> /LGICMD=SYS$MANAGER:CAPTIVE.COM
UAF> EXIT
```

Ensure that the file has the correct protection. Grant group or world access to the file.

Example: Login to a Captive Account

```
Username: SPECULATE
Password:
```

```
      Welcome to SPECULATE
```

```
❶ Speculate> USE INVESTMENT DATA
Spec: Consider it done.
Speculate> COMPUTE FOR NEXT 10 YEARS
SPEC: Please specify commodity.
Speculate> GOLD
Spec: $933,999,456,657.32
Speculate>
.
.
.

(User continues to interact with the SPECULATE program.)

.
.
.
❷ Speculate> BYE
SPECULATE logged out at 04-DEC-1999 16:00:15.16
```

Example Notes: Login to a Captive Account

Notes on the Login to a Captive Account include:

- ❶ Once the user has successfully entered the account password, the special login command procedure immediately executes a data management program.
- ❷ The user can enter commands to this program until the termination command BYE is entered. At that point, the login procedure instructs the operating system to log out the process.

Example: Captive Command Procedure

This command procedure was used in the captive account example.

```
$!CAPTIVE.COM
$!
$ ON ERROR THEN LOGOUT
$ SET NOCONTROL=Y
$ ON CONTROL_Y THEN LOGOUT
$ WRITE SYS$OUTPUT "Welcome to SPECULATE"
$ DEFINE/USER SYS$INPUT SYS$COMMAND
$ RUN SYS$SYSTEM:SPECULATE
$ LOGOUT
```

RESTRICTED and CAPTIVE AUTHORIZE Flags

There is a difference between the AUTHORIZE flags CAPTIVE and RESTRICTED.

/FLAGS=CAPTIVE

- Prevents users from changing any defaults at login.
- Prevents users from escaping from the captive login command procedure to the DCL level.
- Disables the use of the INQUIRE command.
- Disables the use of Ctrl/Y.
- Prevents the use of SPAWN.
 - Unless the SPAWN command carries the /TRUSTED qualifier it is ineffective within a captive account.
 - SPAWN is also disabled from MAIL and the DEC Text Processing Utility (DECTPU) (as a built-in procedure).

/FLAGS=RESTRICTED

- Prevents users from changing any defaults at login.
- Allows the user access to the DCL command level following the execution of the system and process login command procedures.
- Does not disable the use of the INQUIRE command in the system and process login command procedures.

Object Security Profile

Overview

With the exception of files, objects inherit security elements from system-supplied template profiles. The security administrator can modify these templates.

Files have a more complicated inheritance mechanism, one that allows for greater control over the security elements of new objects.

The security elements of any object make up its **security profile**. The object security profile contains:

- **Owner** of the object
- **Protection code**
- **Access control lists (ACLs)**

Topics

In this section the following topics are discussed:

- Ownership and protection of objects
- Access control lists
- Manipulating an object's security profile

Ownership and Protection of Objects

Ownership Element

The first element of an object's security profile is the UIC of the owner. In most cases, the process that creates the object is its owner. The ownership element allows:

- The owner process to modify the security profile
- The system to identify ownership to use in making access decisions

Who Can Change Ownership?

The owner of any object that is not a file can reassign ownership to another user with the SET SECURITY/OWNER command.

A new file obtains its owner from a number of sources. If any of the following conditions are true, then you can assign an identifier as the owner of a file:

- The identifier matches your process UIC.
- You hold the identifier with the Resource attribute.
- You hold GRPPRV privilege and the identifier's group number matches your UIC group.
- You hold SYSPRV privilege.

Source of a File's Owner Identifier

A file receives its owner identifier from the first applicable source that you are allowed to assign:

- The explicit assignment of an owner at creation with the /OWNER_UIC qualifier to the CREATE or COPY command
- The previous version
- The parent directory
- The process UIC

Protection

The second element of an object's security profile is the object's protection code. Protection codes are assigned to objects automatically by the system. Protection codes:

- Determine the type of access allowed to a user
 - Based on the relationship between the user's UIC and the owner UIC
 - Except for files, the code a protected object gets comes from the template profile for the class
- Are relied on if the object is to be accessed by:
 - Only the owner
 - All users on the system
 - A specific UIC-based group of users

An ACL or a privilege can still allow a user to access an object.

Access Control Lists

Why Access Control Lists Are Needed

Occasionally a user in one group may need regular access to a certain file belonging to another group. You could give the SYSPRV privilege to the user desiring access, or you could change the protection on the file to allow World users more access. However, these solutions are too broad for typical systems. Most managers are understandably reluctant to give the powerful SYSPRV privilege to users, and the file owner may not want to allow access to all users.

While OpenVMS system protection using UICs is usually sufficient for most files on your system, the OpenVMS operating system provides an additional layer of protection that you can use to grant access to files for specific users rather than groups of users.

To grant access to specific groups of users outside a single UIC group but not to all users on the system, create an access control list (ACL).

Access Control Lists

The third (optional) element of an object's security profile is the object's access control list. An access control list is a file containing entries that define the access rights that a user or group of users has to a particular protected object, such as a file, directory, or device. ACLs can be created by:

- Default when an object is created
- The security administrator
- Users who have control access to the objects

Determining if an Object Has an ACL

Because ACLs can be placed on files by default or by the system administrator, the user may not be aware that there is an ACL on a file. One way to determine whether there is an ACL on a file is to enter the following command:

```
$ DIRECTORY/SECURITY
```

Displaying an ACL

Use the DCL SHOW SECURITY command to display ACLs. For more extensive ACL work, there is an ACL editor available.

**Note**

The DCL commands SET SECURITY and SHOW SECURITY replace the SET ACL and SHOW ACL commands although the old commands still work as expected.

How Access Is Determined

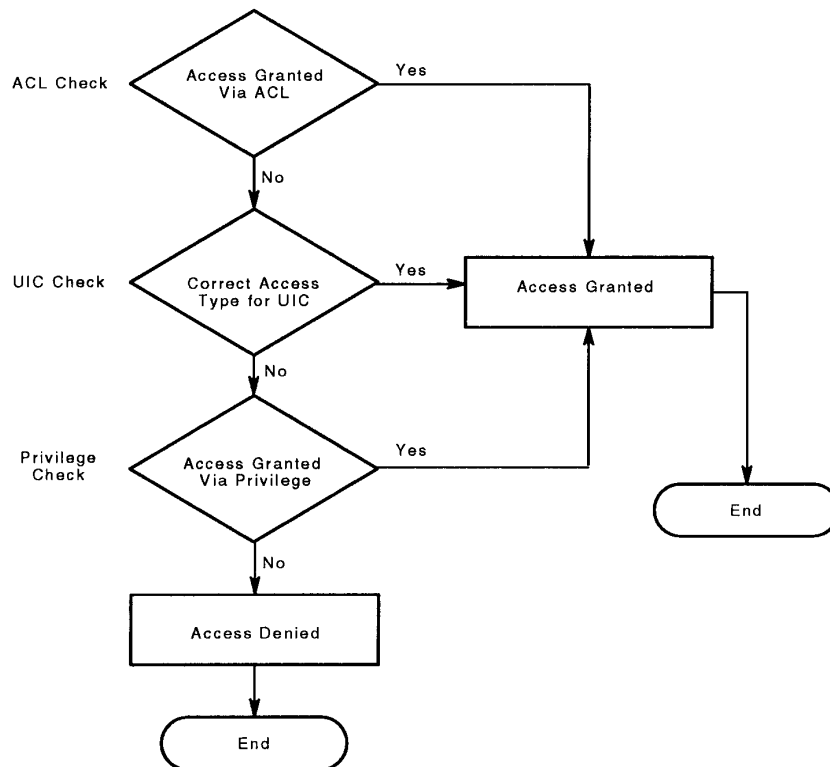
When a user tries to access a protected object, the OpenVMS system compares the security profile for the user process with the security profile of the object.

When the system evaluates a user's right to access an object, it follows this sequence:

Step	Function
1	Evaluate the ACL.
2	Evaluate the protection code.
3	Look for special privileges.
4	Consider access overrides.

Figure: Object Access

The following flowchart shows the steps used in checking object access:



ZKO-055-000057-23-DG

Step 1: ACL Check

If an object has an ACL:

- The system scans it, looking for an Identifier ACE that matches any of the user's right identifiers.
- If a matching ACE is found that grants access:
 - Access is given.
 - Further checking stops.
- If a matching ACE is found that denies access:
- Scanning stops.
- The system evaluates the protection code.
- If no matching ACE is found:
- The system evaluates the protection code.

Step 2: Protection Code Check

If an ACL has *denied* access:

- The system examines the system and owner fields of the protection code.
- The user can still acquire access by:
 - Being a member of the system or owner categories.
 - Possessing privileges.

If access was not *granted* by an ACL:

- And if the object's owner's UIC is not zero:
 - Access is granted or denied based on the relationship between the UIC and the object's protection code.
- And if an object has an owner UIC of zero:
 - The protection code is not checked.
 - Users have all but control access to the object, *provided* the ACL has no Identifier ACEs.
 - If Identifier ACEs are present, access has to be granted through the ACL or through privilege.

Step 3: Privilege Check

If access was not granted by the ACL or the protection code:

- Users with certain system privileges may be entitled to access, regardless of the protection offered by the ACLs or the protection code.
- The bypass privilege (BYPASS), group privilege (GRPPRV), read all privilege (READALL), or system privilege (SYSPRV) amplifies the holder's access to objects.

Step 4: Access Overrides

For some object classes, access may be granted based on alternate privileges. For example:

- The queue object allows full access to all queues for users with operator privilege (OPER).
- The logical name table object allows access to the system table for users with system name privileges (SYSNAM).

Manipulating an Object's Security Profile

SET SECURITY Command

The system manager can:

- Assign new values for the owner, protection code, or ACL of a protected object
- Copy a security profile from one object to another

Perform these functions using the DCL SET SECURITY command.

Before changing an object's security it is important to know:

- Object's current security
- Object's class

Displaying an Object's Security Profile

To display the security profile for any protected object, use the DCL SHOW SECURITY command.

Format:

SHOW SECURITY object-name

Example: Displaying an Object's Security Profile

The following example shows how to display an object's security profile:

```
❶ $ SHOW SECURITY PROJECT_PLAN.TXT

WORK_DISK$:[SARAH]PROJECT_PLAN.TXT;2 object of class FILE
❷ Owner: [SARAH]
❸ Protection: (System: RWED, Owner: RWED, Group: RE, World)
❹ Access Control List: <empty>
```

Example Notes: Displaying an Object's Security Profile

Notes on the Displaying an Object's Security Profile include:

- ❶ The SHOW SECURITY command displays an object's security profile.
- ❷ User Sarah owns the file.
- ❸ The file's protection code gives read, write, execute, and delete access to system users and the owner. The code also grants read and execute access to group users and no access to world users.
- ❹ There is no ACL on the file.

Object Classes

Groups of objects that behave in a particular way and have a common set of security attributes are grouped in classes. The OpenVMS operating system supports 11 classes of protected objects.

File is the default object class for the SET SECURITY command.

Class Name	Description
Capability	Resource to which the system controls access (a vector processor, for example).
Common event flag cluster	Set of 32 event flags that enable cooperating processes to post event notifications to each other.
Device	Class of peripherals connected to a processor, which are capable of receiving, storing, or transmitting data.
File	Files-11 On-Disk Structure Level 2 (ODS-2) files and directories.
Group global section	Shareable memory section potentially available to all processes in the same group.
Logical name table	Shareable table of logical names and their equivalent names for the system or a particular group.
Queue	Set of jobs to be processed in a batch, terminal, server, or print job queue.
Resource domain	Namespace controlling access to OpenVMS distributed lock management resources.
Security class	Data structure containing the elements and management routines for all members of the security class.
System global section	Shareable memory section potentially available to all processes in the system.
Volume	Mass storage medium, such as a disk or tape, that is in ODS-2 format. Volumes contain files and may be mounted on devices.

Example: Changing the Security Profile of a Nonfile Object

When you modify the profile of an object, you need to specify the class of the object. Otherwise the SET SECURITY command assumes the object is a file.

In the following example, the command sequence changes the profile of an object and uses the /CLASS qualifier to identify the object LNM\$GROUP as a logical name table.

```
$ SET SECURITY /CLASS=LOGICAL_NAME_TABLE-
❶ _$ /OWNER=MARKETING /PROTECTION=(S:RWCD, O:RWCD, G:R, W:R)-
❷ _$ /ACL=(( IDENTIFIER=SARAH, ACCESS=CONTROL ), -
❸ _$ ( IDENTIFIER=BONNIE, ACCESS=READ+WRITE )) LNM$GROUP
```

```
❹ $ SHOW SECURITY LNM$GROUP /CLASS=LOGICAL_NAME_TABLE
```

```
LNM$GROUP object of class LOGICAL_NAME_TABLE
```

```
Owner: [MARKETING]
Protection: (System: RWCD, Owner: RWCD, Group: R, World: R)
Access Control List:
  ( IDENTIFIER=[USER, SARAH], ACCESS=CONTROL )
  ( IDENTIFIER=[USER, BONNIE], ACCESS=READ+WRITE )
```

Example Notes: Changing the Security Profile of a Nonfile Object

Notes on the Changing the Security Profile of a Nonfile Object include:

- ❶ Owner of the logical name table was set to Marketing.
- ❷ Protection code was changed to allow read, write, create, and delete access for system users and the owner, but limit group and world users to read access.
- ❸ An ACL is created to allow control access for user Sarah and to allow read and write access for user Bonnie.
- ❹ The SHOW SECURITY command displays the results of the changes.

Required Access for Changing an Object's Security Profile

To modify a security profile, you need control access to the object. An ACL grants control access explicitly; whereas a protection code grants it implicitly to anyone who belongs to the owner or system category.

Example: Modifying an Object's Security Profile

The following example shows how to modify an object's security profile:

```
❶ $ SET SECURITY/PROTECTION=(G:RWED) PROJECT_PLAN.TXT

❷ $ SHOW SECURITY PROJECT_PLAN.TXT

WORK_DISK$:[SARAH]PROJECT_PLAN.TXT;2 object of class FILE
  Owner: [VMS,SARAH]
  Protection: (System: RWED, Owner: RWED, Group: RWED, World)
  Access Control List: <empty>
```

Example Notes: Modifying an Object's Security Profile

Notes on the Modifying an Object's Security Profile include:

- ❶ User Sarah owns the file PROJECT_PLAN.TXT. As owner, she can change the file protection code. Originally, it gave only read and execute access to users in the group category. Using the SET SECURITY command, Sarah changes the protection code to allow read, write, execute, and delete access to group users.
- ❷ The SHOW SECURITY command verifies the new protection code for the file.

Using Passwords to Maintain System Security

Overview

The OpenVMS operating system provides several ways to use passwords to help maintain system security.

Topics

In this section the following topics are discussed:

- Defining user passwords
- Filtering passwords
- Defining a system password
- Examples of password usage

Defining User Passwords

Password Options

Each user is typically assigned a user name and a password in the user authorization file, SYSUAF.DAT. The user must enter both before using the system.

The OpenVMS operating system protects passwords from disclosure through **encryption**. Algorithms transform passwords from plain text strings into cipher text, which is stored in the UAF file. A password check is done on the encrypted password, not the plain text.

The system manager can select any or all three options with user passwords:

- Require a password.
- Require a secondary password.
- Require a system-defined password.

Requiring User Passwords

Use the Authorize utility to set:

- Expiration date on UAF records
- Minimum password length
- Expiration date for password
- Passwords that must be changed when user logs in for the first time

Example: Setting User Passwords

Use the Authorize utility to modify any user password.

```
UAF> MODIFY USER1 /PASSWORD=FIRST
```


Requiring a Secondary Password

Use the Authorize utility to define a secondary password for an account.

```
UAF> MODIFY USER1 /PASSWORD=( "", SECOND)
```

This command does not affect the current value of the primary password if you correctly specify a null first password string.



Note

The DECnet software cannot handle secondary passwords within access strings.

The effectiveness of a secondary password depends on the trustworthiness of the supervisor who supplies it because the supervisor can remove the secondary password by changing it to a null string.

Requiring a Generated Password

Use the /FLAG=GENPWD qualifier within the AUTHORIZE utility to force users to use the password generation facility to set their primary or secondary passwords (or both).

```
UAF> MODIFY/FLAG=GENPWD
```

If this flag has been set in the UAF record, the SET PASSWORD command displays a list of generated passwords for the user.



Note

The minimum number of characters for each generated password comes from the minimum password value in the user's UAF record (set with the AUTHORIZE qualifier /PWDMINIMUM). With the SET PASSWORD/GENERATE command the user must specify a value greater than or equal to this UAF field value.

Filtering Passwords

Filtered passwords are user-selected passwords that have been screened for acceptability. Filtering is *not* done on system generated passwords or passwords set in AUTHORIZE.

Dictionary Search

New passwords are automatically checked against a system password dictionary to make sure the word is not present in the dictionary.

- The system password dictionary is kept in SYS\$LIBRARY.
- /FLAGS=DISPWDDIC is the AUTHORIZE flag placed on user accounts to disable the dictionary search.

History Search

The system keeps a list of the previous passwords used by each user and compares the new password with the list.

- SYS\$SYSTEM:VMS\$PASSWORD_HISTORY.DATA contains the password history list.
- /FLAGS=DISPWDHIS is the AUTHORIZE flag placed on user accounts to disable the history search.
- The system logical name SYS\$PASSWORD_HISTORY_LIMIT establishes the number of passwords on a history list. By default, it is 100.

Defining System Passwords

System Password

For terminals in remote or sensitive locations, the system manager can require a system password. This password is kept as a special UAF record.

Defining a System Password

The system manager seldom executes the SET TERMINAL command interactively. Instead, commands would be placed in either SYCONFIG.COM (preferred) or in the general SYSTARTUP_VMS.COM startup procedures, depending on the preference of the system manager.

Steps for Defining a System Password

To define a system password:

Step	Action	Command
1	Enter the old password in response to the first prompt. Enter the new password in response to the next two prompts. Passwords are not echoed while being entered.	<pre>\$ SET PASSWORD/SYSTEM Old password: New password: Verification: (This command requires SECURITY and CMKRNL privileges.)</pre>
2	Set the specified terminal to require the system password. To log in to this terminal, a user must press the Return key and enter the system password (no prompt is given). If successful, the user continues the normal login procedure when the Username: prompt appears.	<pre>\$ SET TERMINAL - _\$ /SYSPASSWORD /PERMANENT - _\$ TXA2 (This command requires LOG_IO privilege.)</pre>

Changing System Passwords

Use AUTHORIZE to initially set and change the system password:

```
UAF> MODIFY /SYSTEM_PASSWORD=password
```

Change existing system passwords by using the DCL command:

```
$ SET PASSWORD/SYSTEM
```

Examples of Password Usage

Example: Normal Password Login

The system does not echo the password.

```
Username: USER1
Password:
```

Note that you can define the password to be null. If the password has been defined to be null, the user receives no password prompt.

Example: Secondary Password Login

Typically, one person knows the primary password and another person knows the secondary password. This ensures that both must be present for login to occur.

```
Username: USER1
Password:
Password:
```

Example: System Password Login

There is no prompt for the system password. The user must press the BREAK key to begin the login process. Until the user has successfully entered the system password, no Username: prompt appears.

```
[Systempassword]
Username: USER1
Password:
```

Summary

Concepts

General System Security

- Protect the physical security of computer equipment by securing the location of the computer.
- Store media in a safe manner.
- Keep SYSUAF.DAT, RIGHTSLIST.DAT, NETPROXY.DAT, and NET\$PROXY.DAT (VAX systems) up to date.
- Use AUTHORIZE flags and hourly restrictions.
- Insist on nontrivial passwords.
- Do not publicize dialup numbers.
- Do not permit WORLD access to SYSUAF.DAT or RIGHTSLIST.DAT.
- Encourage use of file protection codes and ACLs.
- Restrict user privileges.
- Create accounting reports to check system usage.
- Label disks and tapes in a systematic manner.
- Manage an OpenVMS Cluster system as a single management domain.
- Privileged users on one VAX node can affect the other nodes.
- There is no node-specific file protection by default.
- A network can provide greater security isolation between nodes than an OpenVMS Cluster environment.
- A single system implies that user names, UICs, and access rights are unique throughout the cluster.
- Multiple user authorization files are not recommended.

User Security Profile

- Users with the same UIC group can share files, volumes, and devices.
- Users with the same UIC group can communicate interactively with each other.
- Users with the same UIC can affect or control each other.
- Rights identifiers represent an individual user or group of users.
- Rights identifiers identify users with possibly differing UIC group numbers as a group.
- Rights identifiers include environmental, general, UIC, system node, and facility identifiers.
- The system manager can grant users privileges to allow precise access and control of operating system resources and facilities.
- Default privileges are assigned at login.
- Authorized privileges must be enabled by the user.
- Privilege categories include NONE, NORMAL, GROUP, DEVOUR, SYSTEM, OBJECTS, and ALL.

Restricting User Activities

- UAF record fields allow the system manager to tailor user accounts by setting values for certain fields and flags.
- Access flags control the time of day, day of the week, and mode of access for a user.
- Security fields define access to files and other system objects.
- The system manager can set login flags to restrict user activities.
- Captive accounts limit user access to system software.
- Restricted accounts prevent users from changing defaults set at login.

Object Security Profile

- The first element of an object's security profile is ownership.
- The second element of an object's security profile is protection.
- The third element of an object's security profile is ACLs.
- ACLs can be created by default, the system manager, or the user.
- ACLs are useful for allowing greater access to users outside a specific group UIC.
- Access is determined in the order of: ACL, protections, privilege, and access override.
- Before changing an object's security profile, it is important to know the current security and the object's class.
- To change a security profile, the user must have control access.
- File is the default object class for the SET SECURITY command.

Using Passwords to Maintain System Security

- System managers can require a password for minimum user security.
- System managers can require a secondary password for more user security.
- System managers can require a system password for remote system user security.
- Filtering performs a dictionary check.
- Filtering performs a history check of the last 100 passwords used.
- System managers can assign a system password to terminals in sensitive locations. No prompt is displayed for the system password.
- Users and managers should be aware of password requirements for accounts and terminals.

Commands

User Security Profile

SHOW PROCESS/ALL

Displays types of identifiers.

MODIFY/PRIVILEGES

AUTHORIZE command used to assign authorized privileges.

MODIFY/DEFPRIVILEGES

AUTHORIZE command used to assign default privileges.

Restricting User Activities

MODIFY/PRIMEDAYS

AUTHORIZE command used to set primary days for a user.

MODIFY/ACCESS

AUTHORIZE command used to control user access to the system.

MODIFY/FLAGS

AUTHORIZE command used to modify a user's login flags.

MODIFY/FLAGS=CAPTIVE/LGICMD=file-spec

AUTHORIZE command used to make a captive user account using the specified login command procedure.

MODIFY/FLAGS=RESTRICTED

AUTHORIZE command used to make a restricted user account.

Object Security Profile

SHOW SECURITY

Displays object's security profile.

SET SECURITY

Sets a file's security profile.

Requires access.

SET SECURITY/CLASS

Sets a nonfile object's security profile.

Requires access.

Using Passwords to Maintain System Security

MODIFY/PASSWORD

AUTHORIZE command to add a secondary password to a UAF record. Does not affect the current value of the primary password if you correctly specify a null first password string.

MODIFY/FLAGS=GENPWD

AUTHORIZE command to require generated passwords for user accounts.

MODIFY/FLAGS=DISPWDDIC

AUTHORIZE command to disable password dictionary searching.

MODIFY/FLAGS=DISPWDHIS

AUTHORIZE command to disable password history checking.

MODIFY/SYSTEM_PASSWORD

AUTHORIZE command to change the system password.

SET PASSWORD/SYSTEM

DCL command to change the system password.

SET TERMINAL/SYSPASSWORD/PERMANENT

DCL command to set the specified terminal to require the system password.

Learning Check

Module 9

Questions

Write the letter of the best answer in the space next to each of the following questions.

1. On a VAX system, physical memory is divided into units called:
 - a. Blocks
 - b. Frames
 - c. Pages
 - d. Tables
2. Where does a process get its working set parameters?
 - a. MODPARAMS.DAT
 - b. SYLOGIN.COM
 - c. SYSTARTUP.COM
 - d. SYSUAF.DAT
3. What parameter determines the initial size of a process working set
 - a. WSDEFAULT
 - b. WSEXTENT
 - c. WSINIT
 - d. WSQUOTA
4. True or false: An instruction must be in memory before a process can execute it.
☐ True
☐ False
5. True or false: The goal of swapping is to keep the maximum number of processes in memory.
☐ True
☐ False
6. How many active circuits can an end node have?
 - a. One
 - b. Two
 - c. 24
 - d. No limit

7. A connection to a service is called a:
 - a. Connection
 - b. Service
 - c. Session
 - d. Transport
8. What mode do you use to communicate directly with a server?
 - a. Direct
 - b. Local
 - c. Server
 - d. Service
9. What LATCP command is used to modify LAT information?
 - a. CHANGE
 - b. MODIFY
 - c. RESET
 - d. SET
10. To create a service alias on individual nodes from the DCL prompt, use
 - a. AUTHORIZE
 - b. LATCP
 - c. NCP
 - d. SYSGEN
11. Suppose you create a user directory \$1\$DIA200:[NEILSEN] and forget to specify that user NEILSEN should own it. What command would you use to correct the ownership of the directory?
 - a. SET FILE /OWNER=NEILSEN \$1\$DIA200:[NEILSEN]
 - b. SET FILE /OWNER=NEILSEN \$1\$DIA200:[000000]NEILSEN.DIR
 - c. SET FILE /BY_OWNER=NEILSEN \$1\$DIA200:[000000]NEILSEN.DIR
 - d. SET FILE /BY_OWNER=NEILSEN \$1\$DIA200:NEILSEN
12. What command procedure automates the procedure for registering or amending a license?
 - a. SYS\$UPDATE:VMSINSTAL.COM
 - b. SYS\$UPDATE:VMSLICENSE.COM
 - c. SYS\$SYSTEM:AUTHORIZE.COM
 - d. SYS\$UPDATE:AUTOGEN.COM

13. When do you usually need to install a product's license key?
 - a. After you install the product software
 - b. Before you install the product software
 - c. Before you read the product documentation
 - d. After users begin to use the product
14. If you wanted to know the amount of free space and total space on a disk volume, which command would you use?
 - a. SHOW VOLUME
 - b. SHOW QUOTA
 - c. SHOW DEVICE/FULL
 - d. SHOW ALL
15. A user complains that an application does not work, and the error message is "RMS-E-FLK, file currently locked by another user" (in other words, the file is open and cannot be opened by two processes at once). What command would help you find out what other process has the file locked?
 - a. SHOW DEVICE /FULL device-name
 - b. DIRECTORY /FULL file-spec
 - c. SHOW DEVICE /FILES device-name
 - d. DIRECTORY /OPEN file-spec
16. Which of the following NCP commands manipulates the volatile database?
 - a. PURGE
 - b. DEFINE
 - c. LIST
 - d. CLEAR
17. What logical name designates a systemwide login command procedure?
 - a. SYS\$LOGIN
 - b. SYS\$LOGIN_PROCEDURE
 - c. SYS\$SYLOGIN
 - d. SYS\$SYSTEM

-
18. Using the Authorize utility, which flag would you set to restrict an account to a specific login command procedure?
- CAPTIVE
 - DEFCLI
 - DISCTLY
 - LOCKPWD
19. Suppose you perform the following actions:
- Boot the cluster nodes BARNUM and RNGLNG, which are connected to the CI bus.
 - Install a new disk drive, named \$1\$DUA33:, on the HSC unit.
 - Initialize \$1\$DUA33:.
 - Enter the MOUNT/CLUSTER command on BARNUM to mount \$1\$DUA33:.
 - Boot the node BAILEY, which is also connected to the CI bus.
 - Boot the node HORSE, which is a satellite.
- Which nodes have \$1\$DUA33: mounted after this sequence of actions?
- BARNUM
 - BARNUM, RNGLNG
 - BARNUM, RNGLNG, BAILEY
 - BARNUM, RNGLNG, BAILEY, HORSE
20. Which of the following commands creates a new entry in the permanent network database?
- SET NODE ROGER ADDRESS 2.4
 - SET NODE 2.4 NAME ROGER
 - DEFINE NODE 2.4 NAME ROGER
 - DEFINE ADDRESS 2.4 NAME ROGER
21. Which of the following should be used to upgrade or update system layered product software?
- SYS\$UPDATE
 - VMSINSTAL
 - SYSGEN
 - INSTALL

22. Which of the following commands displays (on one screen) several of the most important classes of information a system manager can monitor?
 - a. MONITOR PROCESSES /TOPCPU
 - b. MONITOR STATES
 - c. MONITOR SYSTEM
 - d. MONITOR ALL_CLASSES
23. What DCL command most quickly shows you which nodes are currently members of the cluster?
 - a. SHOW CLUSTER
 - b. SHOW MEMBERS
 - c. MONITOR NODES
 - d. MONITOR CLUSTER
24. What DCL command most easily shows you what program a process is running?
 - a. SHOW IMAGE
 - b. SHOW USERS
 - c. MONITOR PROCESSES
 - d. SHOW PROCESS/CONTINUOUS
25. A disk device is connected to cluster node DD, which has allocation class 44. The device has unit number 33 and device type DU. Which of the following is a valid name for the device?
 - a. \$DD\$DUA44
 - b. \$44\$DDA33
 - c. \$33\$DUA44
 - d. \$44\$DUA33
26. What file do you edit to make parameter changes permanently known to AUTOGEN?
 - a. SYS\$MANAGER:VMSIMAGES.DAT
 - b. SYS\$UPDATE:AUTOGEN.DAT
 - c. SYS\$SYSTEM:SETPARAMS.DAT
 - d. SYS\$SYSTEM:MODPARAMS.DAT

27. Which DCL command is used to create a queue?
- a. INITIALIZE/QUEUE
 - b. CREATE/QUEUE
 - c. START/QUEUE
 - d. ASSIGN/QUEUE
28. In which file are terminal speeds, queues, and other system management related operations usually specified?
- a. SYLOGIN.COM
 - b. SYSTARTUP_VMS.COM
 - c. STARTUP.COM
 - d. SYLOGICALS.COM
29. Which procedure should mount site-specific volumes?
- a. SYSTARTUP_VMS.COM
 - b. STARTUP.COM
 - c. SYLOGICALS.COM
 - d. SYCONFIG.COM
30. Which procedure should define standard logical names such as SYS\$SYLOGIN, SYS\$ANNOUNCE, and SYS\$WELCOME?
- a. SYLOGICALS.COM
 - b. SYCONFIG.COM
 - c. SYSTARTUP_VMS.COM
 - d. STARTUP.COM
31. Before an active queue can be deleted, it must first be
- a. Initialized
 - b. Stopped
 - c. Spooled
 - d. Started
32. What file controls user privileges and quotas?
- a. NETPROXY.DAT
 - b. QUOTA.SYS
 - c. SYLOGIN.COM
 - d. SYSUAF.DAT

- 33. What do you use to adjust values for system parameters?
 - a. AUTHORIZE
 - b. DISKQUOTA
 - c. SYSGEN
 - d. VMSINSTAL
- 34. Which utility do you use to install a layered product?
 - a. INSTALL
 - b. LAYERED
 - c. SYS\$INSTALL
 - d. VMSINSTAL
- 35. True or false: You must permit WORLD access to SYSUAF.DAT so users can log in.
 - ☐ True
 - ☐ False
- 36. What can you use to grant file access to specific groups of users?
 - a. Access control list
 - b. File access privileges
 - c. Protection codes
 - d. User identification code
- 37. What DCL command is used to change a password?
 - a. CHANGE PASSWORD
 - b. MODIFY PASSWORD
 - c. SET PASSWORD
 - d. None of the above
- 38. What AUTHORIZE flag can be used to require a user to choose a password created by the system?
 - a. CAPTIVE
 - b. GENPWD
 - c. LOCKPWD
 - d. SYSPWD

39. Where does a process get its UIC?
- a. LOGIN.COM
 - b. NETPROXY.DAT
 - c. SYLOGIN.COM
 - d. SYSUAF.DAT
40. What DCL command do you use to display the identifiers for your current process?
- a. SHOW CURRENT
 - b. SHOW IDENTIFIERS
 - c. SHOW PROCESS/ALL
 - d. SHOW RIGHTS/PROCESS
41. What do you use to assign privileges to users?
- a. AUTHORIZE
 - b. AUTOGEN
 - c. MODPARAMS
 - d. SETPRIV

Learning Check Answers

Module 10

1. On a VAX system, physical memory is divided into units called
 - a. Blocks
 - b. Frames
 - c. Pages
 - d. Tables
2. Where does a process get its working set parameters?
 - a. MODPARAMS.DAT
 - b. SYLOGIN.COM
 - c. SYSTARTUP.COM
 - d. SYSUAF.DAT
3. What parameter determines the initial size of a process working set?
 - a. WSDEFAULT
 - b. WSEXTENT
 - c. WSINIT
 - d. WSQUOTA
4. True or false: An instruction must be in memory before a process can execute it.
☐ True
☐ False
5. True or false: The goal of swapping is to keep the maximum number of processes in memory.
☐ True
☐ False

6. How many active circuits can an end node have?
 - a. One
 - b. Two
 - c. 24
 - d. No limit
7. A connection to a service is called a
 - a. Connection
 - b. Service
 - c. Session
 - d. Transport
8. What mode do you use to communicate directly with a server?
 - a. Direct
 - b. Local
 - c. Server
 - d. Service
9. What LATCP command is used to modify LAT information?
 - a. CHANGE
 - b. MODIFY
 - c. RESET
 - d. SET
10. To create a service alias on individual nodes from the DCL prompt, use
 - a. AUTHORIZE
 - b. LATCP
 - c. NCP
 - d. SYSGEN
11. Suppose you create a user directory \$1\$DIA200:[NEILSEN] and forget to specify that user NEILSEN should own it. What command would you use to correct the ownership of the directory?
 - a. SET FILE /OWNER=NEILSEN \$1\$DIA200:[NEILSEN]
 - b. SET FILE /OWNER=NEILSEN \$1\$DIA200:[000000]NEILSEN.DIR
 - c. SET FILE /BY_OWNER=NEILSEN \$1\$DIA200:[000000]NEILSEN.DIR
 - d. SET FILE /BY_OWNER=NEILSEN \$1\$DIA200:NEILSEN

12. What command procedure automates the procedure for registering or amending a license?
 - a. SYS\$UPDATE:VMSINSTAL.COM
 - b. SYS\$UPDATE:VMSLICENSE.COM
 - c. SYS\$SYSTEM:AUTHORIZE.COM
 - d. SYS\$UPDATE:AUTOGEN.COM
13. When do you usually need to install a product's license key?
 - a. After you install the product software
 - b. Before you install the product software
 - c. Before you read the product documentation
 - d. After users begin to use the product
14. If you wanted to know the amount of free space and total space on a disk volume, which command would you use?
 - a. SHOW VOLUME
 - b. SHOW QUOTA
 - c. SHOW DEVICE/FULL
 - d. SHOW ALL
15. A user complains that an application does not work, and the error message is "RMS-E-FLK, file currently locked by another user" (in other words, the file is open and cannot be opened by two processes at once). What command would help you find out what other process has the file locked?
 - a. SHOW DEVICE /FULL device-name
 - b. DIRECTORY /FULL file-spec
 - c. SHOW DEVICE /FILES device-name
 - d. DIRECTORY /OPEN file-spec
16. Which of the following NCP commands manipulates the volatile database?
 - a. PURGE
 - b. DEFINE
 - c. LIST
 - d. CLEAR

17. What logical name designates a systemwide login command procedure?
- a. SYS\$LOGIN
 - b. SYS\$LOGIN_PROCEDURE
 - c. SYS\$SYLOGIN
 - d. SYS\$SYSTEM
18. Using the Authorize utility, which flag would you set to restrict an account to a specific login command procedure?
- a. CAPTIVE
 - b. DEFCLI
 - c. DISCTLY
 - d. LOCKPWD
19. Suppose you perform the following actions:

- Boot the cluster nodes BARNUM and RNGLNG, which are connected to the CI bus.
- Install a new disk drive, named \$1\$DUA33:, on the HSC unit.
- Initialize \$1\$DUA33:.
- Enter the MOUNT/CLUSTER command on BARNUM to mount \$1\$DUA33:.
- Boot the node BAILEY, which is also connected to the CI bus.
- Boot the node HORSE, which is a satellite.

Which nodes have \$1\$DUA33: mounted after this sequence of actions?

- a. BARNUM
 - b. BARNUM, RNGLNG
 - c. BARNUM, RNGLNG, BAILEY
 - d. BARNUM, RNGLNG, BAILEY, HORSE
20. Which of the following commands creates a new entry in the permanent network database?
- a. SET NODE ROGER ADDRESS 2.4
 - b. SET NODE 2.4 NAME ROGER
 - c. DEFINE NODE 2.4 NAME ROGER
 - d. DEFINE ADDRESS 2.4 NAME ROGER

21. Which of the following should be used to upgrade or update system layered product software?
 - a. SYS\$UPDATE
 - b. VMSINSTAL
 - c. SYSGEN
 - d. INSTALL
22. Which of the following commands displays (on one screen) several of the most important classes of information a system manager can monitor?
 - a. MONITOR PROCESSES /TOPCPU
 - b. MONITOR STATES
 - c. MONITOR SYSTEM
 - d. MONITOR ALL_CLASSES
23. What DCL command most quickly shows you which nodes are currently members of the cluster?
 - a. SHOW CLUSTER
 - b. SHOW MEMBERS
 - c. MONITOR NODES
 - d. MONITOR CLUSTER
24. What DCL command most easily shows you what program a process is running?
 - a. SHOW IMAGE
 - b. SHOW USERS
 - c. MONITOR PROCESSES
 - d. SHOW PROCESS/CONTINUOUS
25. A disk device is connected to cluster node DD, which has allocation class 44. The device has unit number 33 and device type DU. Which of the following is a valid name for the device?
 - a. \$DD\$DUA44
 - b. \$44\$DDA33
 - c. \$33\$DUA44
 - d. \$44\$DUA33

26. What file do you edit to make parameter changes permanently known to AUTOGEN?
- a. SYS\$MANAGER:VMSIMAGES.DAT
 - b. SYS\$UPDATE:AUTOGEN.DAT
 - c. SYS\$SYSTEM:SETPARAMS.DAT
 - d. SYS\$SYSTEM:MODPARAMS.DAT
27. Which DCL command is used to create a queue?
- a. INITIALIZE/QUEUE
 - b. CREATE/QUEUE
 - c. START/QUEUE
 - d. ASSIGN/QUEUE
28. In which file are terminal speeds, queues, and other system management related operations usually specified?
- a. SYLOGIN.COM
 - b. SYSTARTUP VMS.COM
 - c. STARTUP.COM
 - d. SYLOGICALS.COM
29. Which procedure should mount site-specific volumes?
- a. SYSTARTUP VMS.COM
 - b. STARTUP.COM
 - c. SYLOGICALS.COM
 - d. SYCONFIG.COM
30. Which procedure should define standard logical names such as SYS\$SYLOGIN, SYS\$ANNOUNCE, and SYS\$WELCOME?
- a. SYLOGICALS.COM
 - b. SYCONFIG.COM
 - c. SYSTARTUP_VMS.COM
 - d. STARTUP.COM
31. Before an active queue can be deleted, it must first be
- a. Initialized
 - b. Stopped
 - c. Spooled
 - d. Started

32. What file controls user privileges and quotas?
- a. NETPROXY.DAT
 - b. QUOTA.SYS
 - c. SYLOGIN.COM
 - d. SYSUAF.DAT
33. What do you use to adjust values for system parameters?
- a. AUTHORIZE
 - b. DISKQUOTA
 - c. SYSGEN
 - d. VMSINSTAL
34. Which utility do you use to install a layered product?
- a. INSTALL
 - b. LAYERED
 - c. SYS\$INSTALL
 - d. VMSINSTAL
35. True or false: You must permit WORLD access to SYSUAF.DAT so users can log in.
- ☐ True
 - ☐ False
36. What can you use to grant file access to specific groups of users?
- a. Access control list
 - b. File access privileges
 - c. Protection codes
 - d. User identification code
37. What DCL command is used to change a password?
- a. CHANGE PASSWORD
 - b. MODIFY PASSWORD
 - c. SET PASSWORD
 - d. None of the above

38. What AUTHORIZE flag can be used to require a user to choose a password created by the system?
- a. CAPTIVE
 - b. GENPWD
 - c. LOCKPWD
 - d. SYSPWD
39. Where does a process get its UIC?
- a. LOGIN.COM
 - b. NETPROXY.DAT
 - c. SYLOGIN.COM
 - d. SYSUAF.DAT
40. What DCL command do you use to display the identifiers for your current process?
- a. SHOW CURRENT
 - b. SHOW IDENTIFIERS
 - c. SHOW PROCESS/ALL
 - d. SHOW RIGHTS/PROCESS
41. What do you use to assign privileges to users?
- a. AUTHORIZE
 - b. AUTOGEN
 - c. MODPARAMS
 - d. SETPRIV

access control list

A list that defines the kinds of access to be granted or denied to users of an object. Access control lists can be created for objects such as files, devices, and mailboxes. Each ACL consists of one or more entries known as access control entries (ACEs).

activity license

Defines the allowed number of concurrent uses of a product.

adjacent nodes

Nodes that have a line connecting them.

alias

A service name that equates to one of several cluster nodes.

area

Subgrouping of nodes in a network.

availability license

Makes a product available to all the users of a system.

baseline

Data that represent the performance of the system when it is functioning normally, under a normal work load.

boot node, boot server

A node that downline loads the operating system software to satellite nodes.

bridge

A device for connecting two segments of a LAN.

captive account

A user account that permits only limited access to software.

circuit

The logical channel between adjacent nodes.

circuit cost

A positive integer associated with using a circuit.

device code

Device type specifier.

device control library

A text library that contains modules that direct printers to perform certain functions.

device name

Unique identifier for each device.

dictionary search

User-selected passwords are checked to make sure they do not appear in the system password dictionary.

disk quotas

Disk restriction management based on UICs.

dump file

Saves a partial copy of physical memory when the system fails.

encryption

A process of encoding information so that its content is no longer immediately obvious to anyone who obtains a copy of it.

execution queue

A queue that accepts batch or print jobs for processing.

filtered passwords

User-selected passwords that have been screened for acceptability.

free page list

A list of pages that are available to processes for allocation.

full name

A hierarchically structured DECnet node name that can be stored in a DECdns naming service.

gateway

A module, or set of modules, that transforms the conventions of one network into the conventions of another.

generated passwords

The OpenVMS operating system provides an automatic password generator that creates randomly generated passwords. This avoids any obvious pattern when assigning passwords.

generic queue

A queue that holds jobs until they can be transferred to an execution queue for processing.

global page

Page that is shared among multiple processes.

global page table

The page table containing the master page table entries for global sections.

global section

A data structure or shareable image section potentially available to all processes in the system.

hard fault

Page fault that requires disk I/O to obtain the virtual page.

hardware controller number

Device controller specifier.

history search

The system maintains a list of the last 100 passwords used by each user. When a user selects a new password it is checked against the user's history list.

hop

The logical distance to an adjacent node.

LAN

Local area network.

LAT network

Local area transport. A collection of nodes connected by Ethernet for the purpose of sharing resources.

license database

A collection of interrelated data stored on a disk and accessed through the LMF software. Each record in the database corresponds to a license.

line

A physical channel between nodes.

local mode

Terminal server operating mode in which an interactive user communicates with a server.

modified page list

List of pages that have been modified since being read from disk.

network

An organized system of two or more computers that are linked together either locally or remotely to communicate and share information.

node

A computer system that communicates with other devices in a network.

object class

A set of protected objects with common elements. For example, all files belong to the file class; all devices belong to the device class.

object security profile

A set of security elements that define access requirements. The elements include an owner (UIC), a UIC-based protection code, and possibly, an ACL.

page

On a VAX system, a contiguous 512-byte portion of memory.

On an AXP system, a page size will be one of four values: 8 kilobytes (8192 bytes), 16 KB, 32 KB, or 64 KB.

page fault

Event that occurs when a process refers to a page that is not in its working set.

page file

System file to which modified pages are written.

pagelet

On an AXP system, a 512-byte quantity.

paging

The action of bringing pages into the working set of a process.

parameter file

This disk file contains the starting values for the system parameters. This file is used during the bootstrap process to initialize the active system parameter values in memory.

path

A possible route from the source node to the destination node.

path cost

The sum of the circuit costs along a path between two nodes.

path length

The total number of hops along a path between two nodes.

path splitting

The ability of DECnet software to divide the load over multiple paths, if multiple paths exist between a pair of nodes.

personal use license

Makes a product available to a set of named users of a system.

port

Connection between a device and a terminal server.

private volume

A volume (disk or tape) owned by a single user or group of users.

product authorization key (PAK)

Contains essential licensing information that you must register in the license database to use the product.

protected object

An object containing shareable information to which the system controls access.

protocol

A series of communications rules implemented with hardware and software.

public volume

A volume (disk or tape) to which all users on the system have access.

quantum

Time slice used in scheduling CPU usage.

quota files

One file per disk that contains quota entries and is managed by the SYSMAN utility.

recording file

A file produced by the Monitor utility, containing MONITOR data in binary form.

remote print queue

A print queue associated with a printer connected to a terminal server.

repeater

Network component used to refresh and regenerate signals.

rights database

The collection of data the system maintains and uses to define and associate identifiers with the holders of the identifiers.

rights list

The list associated with each process that includes all the identifiers the process holds.

routing node

A network node that forwards information intended for other nodes.

satellite node

A node that does not have a local system disk, and must load the operating system software from a boot server.

secondary password

A user password that may be required at login time immediately after the primary password has been correctly entered. Primary and secondary passwords can be known by separate users to ensure that more than one user is present at the login.

security profile

A set of elements that describe either an object's access requirements or a subject's access rights.

server

An entity that is responsible for processing requests.

service

A name associated with an application on a port or a node.

service mode

Terminal server operating mode in which an interactive user communicates with a service.

session

A connection to a service.

soft fault

Page fault that can be satisfied without performing disk I/O.

spooled device

A device that directs the output of an application to an intermediate file until the application program finishes.

summary file

A file produced by the Monitor utility, containing a summary of selected data in text form.

swap file

Manages physical memory use.

swapping

The operation of writing a process working set out to a disk file known as the swap file.

symbiont

A process that transfers data between a disk and an input or output device.

system integrated products (SIPs)

Layered software products that are included as part of the OpenVMS kit, which require no further installation beyond registering a PAK.

system parameters

The values that control such things as the sizes of the OpenVMS data structures in memory, the number of installed image files, and the sizes of the system files.

system password

A password required by a terminal before login can be initiated.

system password dictionary

A collection of unacceptable words for filtering user-selected passwords.

system startup files

A series of command procedures that are invoked when the system is started up.

terminal server

Component on a local area network that connects devices such as terminals, printers, modems, and computers to the network.

unit number

Indicates position of the device on the controller.

virtual memory management system

Operating system that allows the execution of programs or images that may not be completely in physical memory.

volume label

An alphanumeric name encoded on a volume.

volume set

Two or more disk volumes bound with the MOUNT/BIND command.

WAN

Wide area network.

working set

The set of physical pages currently being used by a process.